Design • Build • Program • Learn • Compete

SERVO MAGAZINE

FOR THE ROBOT INNOVATOR

www.servomagazine.com

December 2013

Robot Builder's Buyer's Guide

Wheels, Motors, & Speed Controllers


Pololu
Robotics & Electronics
Finding the right parts for your robot can be difficult, but you also don't want to spend all your time reinventing the wheel (or motor controller). That's where we come in: Pololu has the unique products — from actuators to wireless modules — that can help you take your robot from idea to reality.
3pi Robot
ITEM #975
$99.95
The 3pi is a great first robot for ambitious beginners and a perfect second robot for those looking to move up from non-programmable or slower beginner robots.
Stepper Motor Drivers
ITEM #2133
$11.95 (qty. 5)
DRV8825 driver
1/32 microstepping
4-layer PCB
ITEM #2128
$9.95 (qty. 5)
A4988 driver
4-layer PCB
Build a 3D printer!
AltIMU-10 Gyro, Accelerometer, Compass, and Altimeter
ITEM #1269
$49.95
Provides ten pressure, rotation, acceleration, and magnetic measurements that can be used to calculate absolute orientation and altitude.
Stocking Stuffers!
Mini Plastic Gearmotor
ITEM #1125
$5.49
Other shapes and gear ratios available
Specs at 4.5 V:
120 RPM
20 oz·in
Micro Maestro USB Servo Controller
ITEM #1350
$19.95
USB, serial, and internal scripting control
12-, 18-, and 24-channel versions available
Continuous Rotation Hobby Servo
ITEM #1248
$12.95
Many other servos available
Specs at 6 V:
54 RPM
71 oz·in
37D mm Metal Gearmotor with 64 CPR Encoder
ITEM #1442
$39.95
Available in six gear ratios, with or without encoder
Specs at 12 V:
500 RPM
84 oz·in
Step-Up/Step-Down Voltage Regulator S7V8A
ITEM #2118
$5.95
2.7 V to 11.8 V input
Adjustable output above or below input voltage
90% efficiency (typical)
Wixel USB Wireless Module
ITEM #1337
$19.95
Do more wire less!
Programmable MCU module featuring a 2.4 GHz radio and USB: write your own software or load precompiled, open-source apps.
Pololu Wheels
STARTING AT
$4.98/pair
Zumo Robot for Arduino
(Assembled with 75:1 HP Motors)
ITEM #2506
$99.95
98 mm
98 mm
Arduino-controllable tracked robot small enough for mini-sumo (less than 10 cm × 10 cm) and flexible enough for you to make it your own.
Includes:
Two micro metal gearmotors coupled to a pair of silicone tracks
Stainless steel bulldozer-style blade
Array of six infrared reflectance sensors for line following or edge detection
3-axis accelerometer and magnetometer
Buzzer for simple sounds and music
Individual parts and kit version also available — build your own configuration!
Build and customize your Zumo!
Simple Motor Controller 18v15
ITEM #1377
$39.95
Advanced motor control made simple
Highly configurable DC motor controller that supports four interface modes: USB, TTL serial, analog voltage, and hobby radio control (RC).
Find these products and more at: www.pololu.com

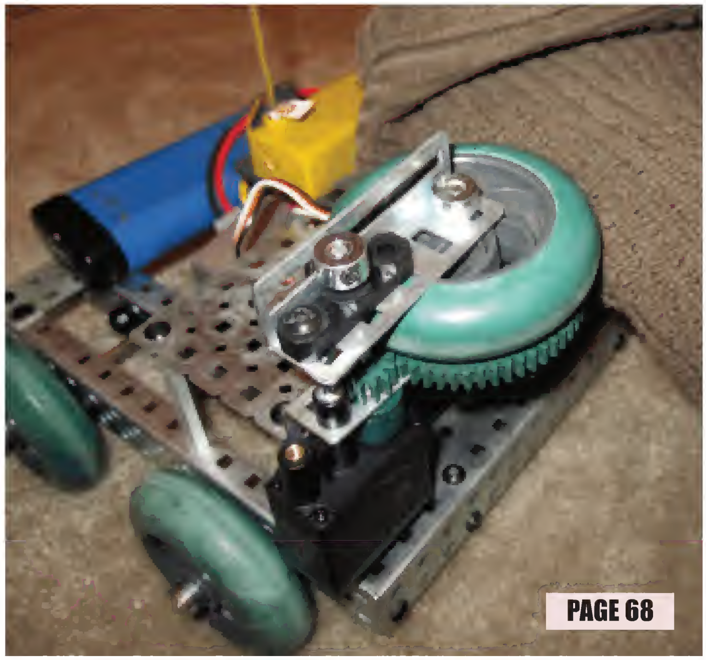
SERVO MAGAZINE

12.2013

VOL. 11 NO. 12

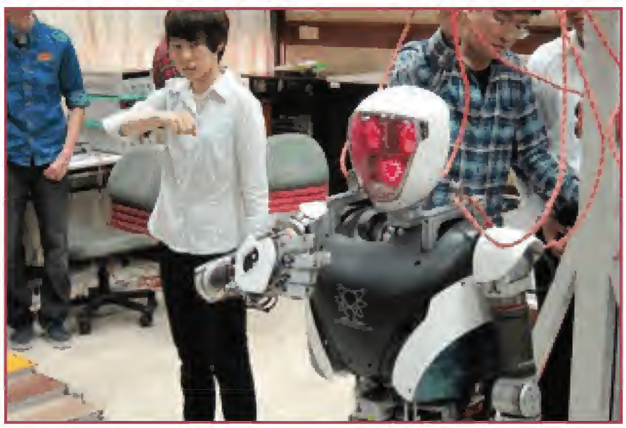
Columns

The Combat Zone...



PAGE 68

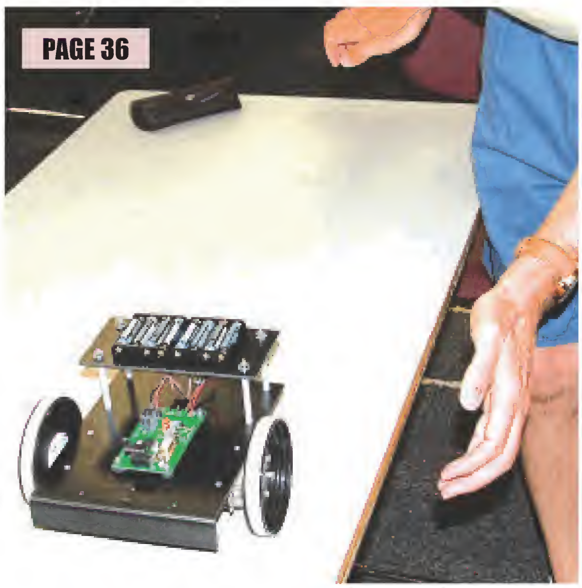
Departments



SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...

PAGE 42



PAGE 36

Mind / Iron

by Bryan Bergeron, Editor

Predicting the Future

Using fMRI (functional magnetic resonance imaging) to view blood flow in the brain, medical researchers can determine how someone will react to a stimulus before that person is consciously aware of their reaction. That is, our brain pre-computes the necessary neuromuscular signals to, say, avoid a slap to the wrist before we consciously move our hand to avoid that slap. In this way, we subconsciously predict the future. This makes sense, if you think about it. Because the propagation of signals in our biochemical communications network is orders of magnitude slower than electrical signals in a length of copper wire, our movements would be erratic and clumsy without some preparation time.

Long before these discoveries were made about the human brain, roboticists were using various methods of predicting the future so that the movements of their robots could be smooth, precise, and executed in the proper order. Consider the Proportional, Integral, Derivative (PID) controller. The three components of PID control can be combined in various ways, with different emphasis on each component to predict the future speed of a basic carpet roamer or velocity of a military drone.

Depending on the application, the three components of PID control can be combined in various ways and with different emphasis on each component. That is, a Proportional (P) controller might be sufficient for a carpet roamer, while a Proportional-Integral (PI) controller might be best for a quadcopter. Proportional control — when used with the appropriate proportion constant and set point correction — can provide much smoother speed. A major limitation of a P controller is an initial period of instability, where predictions about future speed vary widely.

A PI control addresses the instability of the purely proportional control scheme by adding a term that represents the sum of the difference between desired and actual speed over time. Because the integral term reflects the difference between actual and desired speed over time, proportional integral control predicts future

Continued on page 34

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER
Robin Lemieux
robin@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, Kevin Berry, Dennis Clark, R. Steven Rainwater, Michael Simpson, Bryce Woolley, Evan Woolley, Morgan Berry, Mike Jeffries, Matt Spurk, John Blankenship, Samuel Mishal, Dan Albert, Paul Verhage

ADVERTISING SALES EXECUTIVE
Richard Lockhart
richard@servomagazine.com

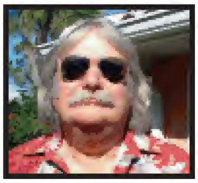
CIRCULATION DEPARTMENT
subscribe@servomagazine.com

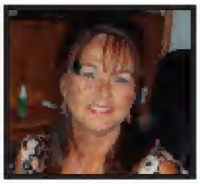
MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION
Sean Lemieux





Robytes

by Jeff and Jenn Eckert

So Long, Casey Jones

Experimental driverless ore truck.

Although Americans are more inclined to think of wool and Foster's as the main Australian imports, our friends "down under" have been making big bucks exporting iron ore since the 1960s and 1970s — particularly to China. In fact, three Aussie companies (Rio Tinto Group, Vale, and BHP Billiton) control nearly 60 percent of the $145 billion global market. Much of the ore is moved by trains which naturally are operated by engineers. Hauling ore from remote mines to seaports is not exactly pleasant work but the mining boom — along with pressure from the Construction, Forestry, Mining, and Energy Union — has gradually pushed about 400 engineer's salaries up to a whopping $224,000 per year. Apparently, the mining companies feel like that's a push too far and have decided to use robotics to solve the problem.

Mining giant Rio Tinto, for example, has allocated a reported $7.2 billion to upgrade its operations, including $518 million to build the world's first fully automated heavy-haul rail system. The rail will have 930 miles of track and 10,000 ore wagons, and each train will stretch over nearly 1.5 miles. Vale is taking a different tack by replacing a large portion of its network with a 23 mile system of conveyor belts. BHP is conducting a trial of 12 driverless trucks. Reportedly, industry regulators in the USA and Canada are somewhat resistant to this level of automation, particularly in light of the runaway train carrying crude oil that derailed and exploded in Quebec last July, killing 47 people and destroying 30 buildings. That train was under the operation of a single engineer who had parked it and left for the night. Even a spokesman for the Aussie union had to acknowledge that "you're never going to win the argument against technology."

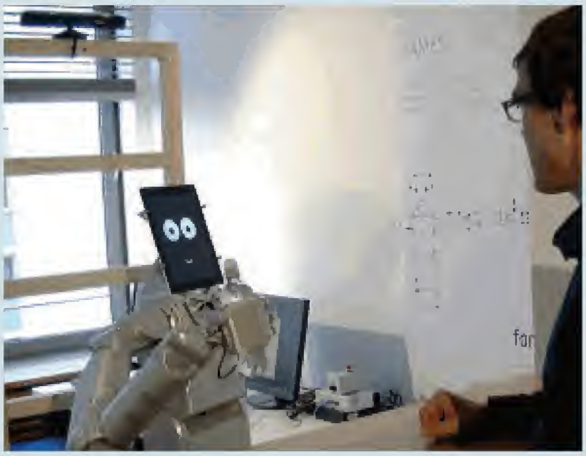
Pickerbot sports stereo vision to judge ripeness and establish target location.

Bye, Bye, Berry Pickers

Strawberry pickers aren't paid on the level of Aussie railroad engineers (but neither are most thoracic surgeons), but they can rake in between $10 and $15 per hour, according to **strawberryplants.org**. But maybe not for long. Japanese agricultural machinery maker Shibuya Seiki (**www.shibuya-sss.co.jp**) has teamed up with the National Agriculture and Food Research Organisation (**www.naro.affrc.go.jp**) to develop a robot that moves on rails between rows of berries (which tend to be grown in greenhouses in Japan) and snatches one off the plants every eight seconds. The bot uses two digital cameras to judge which ones are ripe (and to calculate the distance to the objective), snaps a high-res photo of each berry, and finally snips it off and drops it into a basket. Eight seconds per strawberry may not sound all that impressive, but that comes to 6.4 minutes for a quart of medium-sized ones, as opposed to 15 minutes per quart for a relatively inexperienced human picker. Plus, the bot can pick continuously — night and day. The system will be available early next year for about $50,000.

Go to **www.servomagazine.com/index.php/magazine/article/december2013_Robytes** to comment on these topics.

Cheers!



Bartender JAMES reads your body language and knows when you want another drink.

The JAMES project at Germany's Bielefeld University (**www.uni-bielefeld.de**) calls to mind Mason Cooley's semi-famous observation, "An academic dialect is perfected when its terms are hard to understand and refer only to one another." JAMES (which stands for Joint Action for Multimodal Embodied Systems) was created by the school's Psycholinguistics Research Group to develop an artificial embodied agent that supports socially appropriate, multi-party, multimodal interaction. JAMES focuses on the qualitative aspects of task achievement in social situations, and how such tasks can be improved through multimodal communication, rather than the physical aspects of traditional robotics tasks." What? Well, project literature further elucidates, "In particular, JAMES plans to develop the core cognitive capabilities that enable a robot to interact with humans in a socially appropriate manner, and demonstrate this behavior in a bartending scenario."

Yeah, it's another tablet-headed, one-armed robotic bartender *and* a clever scheme to get academic credit for drinking beer.

Art for Art's Sake

A robot reproduces a drawing from afar.

Finally, this month's Big Whoop Award goes to Austrian artist Alex Kiessling (**www.alexkiessling.com**) who aspired to astonish the universe by drawing three pictures at once last September. While he worked on the original in Vienna, a pair of industrial robots — one in London's Trafalgar Square and the other in Berlin's Breitscheidplatz — drew the same picture using satellite data transmission. Spectators in all three locations could follow the redundant activities on giant screens. Sadly, a photo in Britain's *Daily Mail* revealed a gathering of perhaps 30 people — half of whom were looking at a nearby fountain instead. Oh, well. **SV**

ASK MR. ROBOTO

by Dennis Clark

Our resident expert on all things robotic is merely an email away.

roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

Post comments on this article at **www.servomagazine.com/index.php/magazine/article/december2013_MrRoboto.**

As I write this, I have just returned from MileHiCon 45. This is an annual Fantasy, Science Fiction, writers, illustrators, and everything else convention held in Denver, CO. Lots of stuff happens there, but that which is of most interest to me is the Critter Crunch. The Critter Crunch is the brainchild of Bill Llewellin of the Denver Mad Scientists Club (of which yours truly is also a member). I'm sure that you've all heard of Robot Wars and BattleBots from TV, and their respective competitions. Critter Crunch is older by far than all of them. In fact, 2013 marks the 24th year of the Critter Crunch. This competition is broken (now) into two brackets: the 2 lb and 20 lb classes. Of these two, the 2 lb is the most competitive; this year, we had 12 entries. In the early years, the classes were dominated by 20/30/40 somethings that were mild mannered (or not) engineers and scientists in their regular lives. Something about mechanical mayhem appealed to these mad scientists. (Perhaps because many of them worked in the defense industry? Dunno ...) Regardless, it is great fun and inexpensive to enter. I have seen everything from cheap RC trucks with custom metal shells to custom-machined works of art compete and win. This year, we saw that this crunchy fun isn't just for the adult engineers and scientists. We had an assortment of adults as usual, but the field was dominated by kids from the age of nine on up to high school students. It's working! Something is pulling the younger generation into engineering remote-controlled battling critters!

The rules of the competition disallow any weapons that would hurl nasty pieces of critters into the unprotected audience since the Critter Crunch does not use an armored protective arena. It is simply an eight foot by eight foot OSB (Oriented Strand Board) surface held off the ground by a simple 2x4 frame sitting on the floor (**Figure 1**). The competition is similar to the familiar Sumo events since you win by forcing your opponent off the combat surface or causing your opponent to be immobile. For the complete rules, look at **www.milehicon.org/Files/Critter%20Crunch%20rules.docx**.

A few years ago, I challenged these "proto" makers to create genuine robots to compete against the humans. Subsequent years have seen an assortment ranging from LEGO RCX-based robots, to microcontroller powered, to an imitation of Grey Walter's Tortoise on the field. I have fielded Silver Surfer for the last three years — each year with slightly improved programming and sensor packages (see **Figure 2**). It has gotten better over time, but I'm no doubt hampered by my tradition of starting work on it a week before the competition. (I am a professional procrastinator.)

A custom robot board whose processor is an Atmel ATMEGA8535 controls Silver Surfer. It is programmed using avr-gcc. It is connected to a laptop via a Bluetooth virtual com port using a SparkFun Bluesmirf board. It has an IR range finder facing forward, and IR proximity detectors on each side and the rear. Finally, the robot uses two downfacing IR proximity detectors to see the edge of the board.

This year, sadly, it went too fast without enough traction and had some issues staying on the board. (Sigh, maybe next year ...) This critter is fully autonomous. I use the terminal connection to calibrate the board edge detectors, as well as start and stop the beast.

I am, of course, happy with the turnout of a new generation of "critterers" — especially since my son, Brendan has designed (with a little help from Mr. Roboto senior) his own bot which he has competed with the last three years. This year, he just missed a trophy; he battled and lost a tie for third place. His critter's name is Munch shown in **Figure 3**.

The only other fully autonomous critter that always attends is Otto whose owner — John of Sussex of England — sends each year to


Figure 1.


Figure 2.

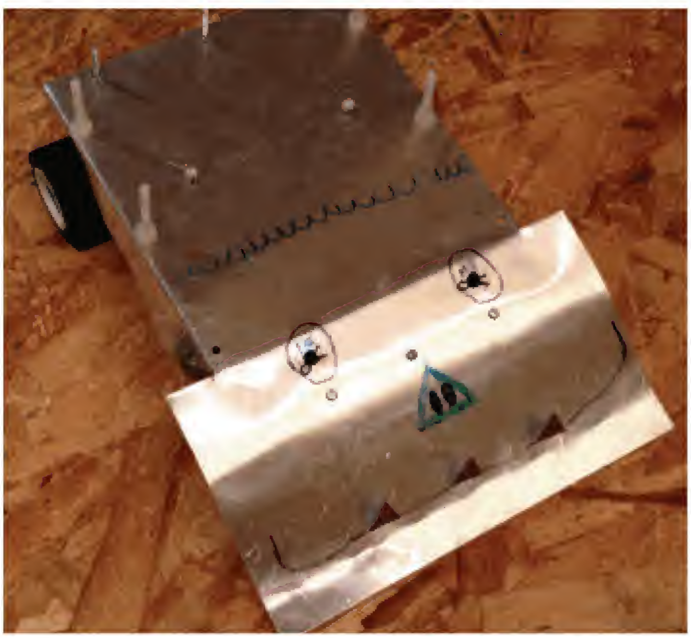


Figure 3.

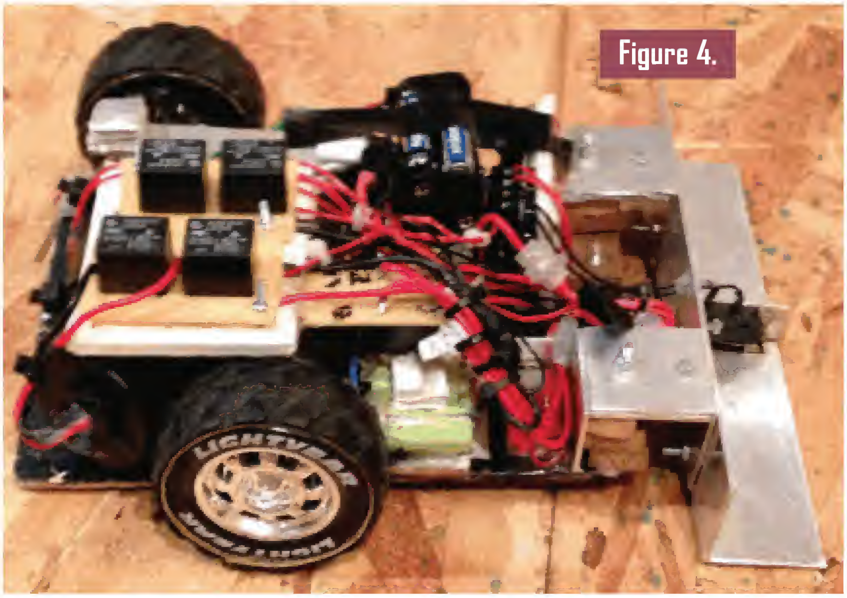


Figure 4.

Figure 5.

Figure 6.

Figure 7.

compete (**Figure 4**). Otto is a minimalist creation with a repertoire of random actions that is sometimes successful.

After the regular competition, the traditional Grand Melee ensues where all of the entries go at the same time in a fierce "every critter for himself" dust-up until only a single bot is left. The audience loves this part.

This year (as is often the case), *SERVO Magazine* donated prizes to the first and second place winners of the competition. Those winners were:

- First Place 2 lb class: Erik Gaalema and his critter, T-Bone (**Figure 5**).
- Second Place 2 lb class: Kathy Bonnet (the one in red) and her builder Ed Doherty and support staff with Bradley (**Figure 6**).
- Third Place 2 lb class: Steve Gaalema and his critter A-Wing (**Figure 7**).

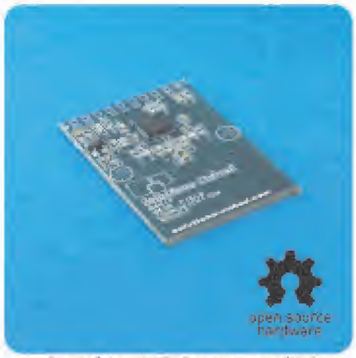
I couldn't stay for the 20 lb class, so I can't report on how that came out.

As usual, if you have any questions about robotics, please send them to roboto@servomagazine.com and I'll do my best to answer them. Until next month, keep building robots. You may earn favor with our future robot overlords if they know that you have! **SV**

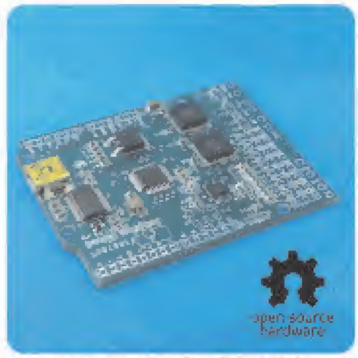
Try Our

Open Source Hardware Modules

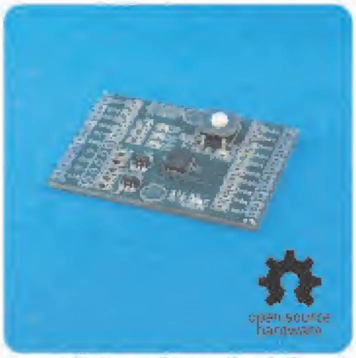
In Your Next Design



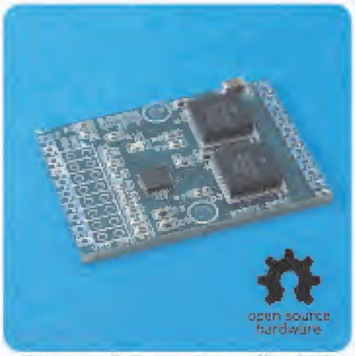
Serial to NFC Converter $16



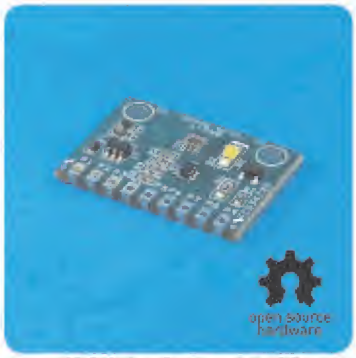
Arduino Compatible With Motor Drivers $35



10 Servo Controller $18

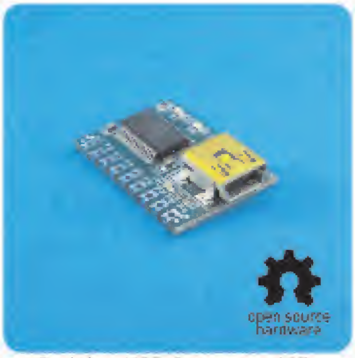


2-Motor 4-Servo Controller $25



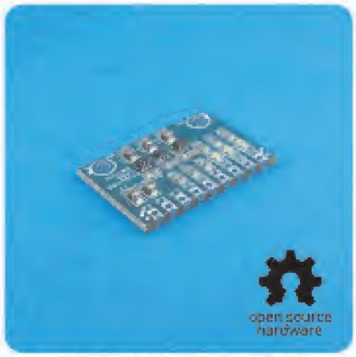
I2C Color Sensor $14.50

IR Object Detector $10

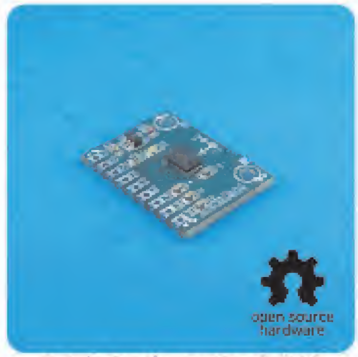


Serial to USB Converter $15

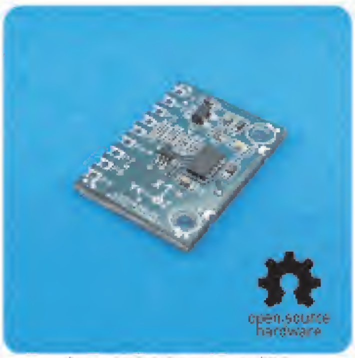
Motor Controller $15



Triple Power Switch $13



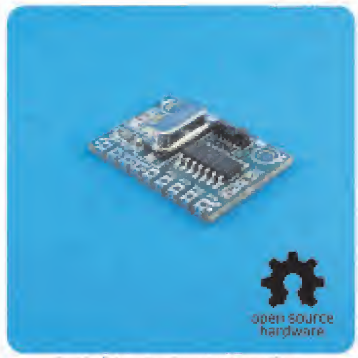
3-Axis Accelerometer $12.50



Electronic Compass $18

PID Motor Controller $30

Clock and Calendar $12

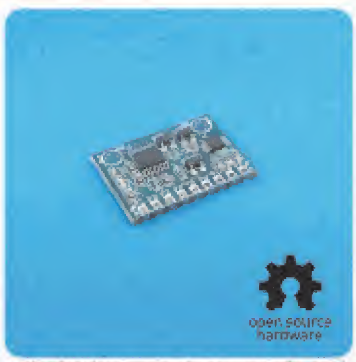


Serial to IR Converter $15

hardware made easy

Solutions Cubed

designservices@solutions-cubed.com
www.solutions-cubed.com



RS232-RS485-TTL Converter $14.50

64Mbit SPI Flash Memory $14

EVENTS

ROBOTS.NET

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**.

— R. Steven Rainwater

DECEMBER

5-8 **ROBOEXOTICA**
Vienna, Austria
The famous annual competition of bartending robots including events for mixing cocktails, server cocktails, lighting cigarettes, bartending conversation, and "other achievements in electronic cocktail culture."
www.roboexotica.org

7-8 **South's BEST Competition**
Auburn University, Auburn, AL
Regional event for the BEST student robotics competition.
www.southsbest.org

16-19 **IROC International Robot Olympiad**
Denver, CO
Events at this year's Olympiad will focus on agricultural robots.
www.iroc.org

JANUARY

3-5 **Techfest**
Indian Institute of Technology, Bombay, India
University level competition for autonomous robots including events like the International Robotics Challenge, Robotron, and Code Master.
www.techfest.org

22-23 **Singapore Robotic Games**
Republic of Singapore
A wide range of events for autonomous robots including Picomouse, Sumo, robot soccer, wall climbing, pole balancing, underwater robots, legged robot marathon, Robot Colony, and an event for humanoid robots.
http://guppy.mpe.nus.edu.sg/srg

23-26 **ION Autonomous Snowplow Competition**
St. Paul, MN
Autonomous snowplow robots compete to remove snow on a designated path.
www.autosnowplow.com

23 **Powered by Sun**
Ostrava, Czech Republic
Autonomous solar-powered robots compete in a race.
http://napajenisluncem.vsb.cz/

31 **Robotix**
IIT Khargpur, West Bengal, India
Large tech competition for university students that includes a variety of robot-related events.
www.robotix.in


THEY SAY THE FUTURE DOESN'T FIT IN A BOX.
Tuesday, January 7
through Friday, January 10, 2014
Las Vegas, Nevada • CESweb.org • #CES2014
GOOD THING LAS VEGAS IS FAR FROM SQUARE.
Over four days, those who shape the future gather in a city built on reinvention. Here, brands, markets and economies converge in what's far more than a tradeshow. And in 2014, there's more opportunity than ever to connect with those who matter. The only question now, why wouldn't you fit it in your future?
Register today at CESweb.org.
2014 International CES
THE GLOBAL STAGE FOR INNOVATION
PRODUCED BY CEA

NEW PRODUCTS

PVC Pipe Clamps

ServoCity has recently released two new PVC pipe clamps to expand their line of Actobotics™ mechanical components. The standardized outside diameter of 1" PVC pipe means that you can go to any hardware store and pick up the length you need with assurance that it will fit into the new pipe clamps. The low cost of PVC pipe allows you to expand the overall size of your project quickly while adding very little cost. The PVC clamp mount (#585524) slides over the PVC pipe and clamps with a single pinch bolt. The 6-32 tapped holes align with the side of the 1.5" Actobotics hub pattern which come in perpendicular to the PVC pipe. The PVC Clamp Hub B (#545512) is an inline clamp that fastens tightly to the PVC pipe using two pinch bolts. The end of the PVC clamp hub B has four 6-32 tapped holes in the 1.5" Actobotics hub pattern. Both of the new PVC clamps are machined from 6061-T6 aluminum. Prices start at $6.99.

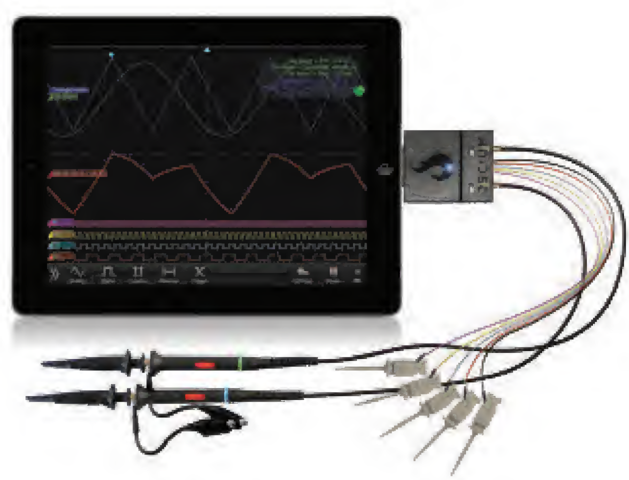
XL Belt Mount

Also available from ServoCity is their new XL Belt Mount to use with their line of XL series timing belts to create a precise drive system. The XL belt mount is constructed of 316 stainless steel and possesses both tapped holes and thru-holes that match up with the .770" Actobotics hub pattern. The belt mount has a 3/8" wide machined groove that properly aligns and tightly holds an XL series timing belt. This mount can be used with a continuous belt circle to fasten a dolly to the belt in a conveyer setup, or it can be used to attach two ends of an open-ended belt. The open cutout in the center of the belt mount allows the ends of an open-ended timing belt to protrude through the middle, which allows the user to adjust belt tension with ease. Price is $3.99.

For further information, please contact:

ServoCity	Website: **www.servocity.com**

OS Oscilloscope



Oscium's newest product — the iMSO-204 — transforms the iPad, iPhone, and iPod touch into an ultra-portable two-channel oscilloscope. iMSO-204 is the second generation of handheld engineering tools from Oscium. With iMSO-204, Oscium has improved both functionality and performance, making the iPad or iPhone a very compelling oscilloscope solution.

Improvements to the original scope include adding a second analog channel and increasing the sample rate by more than a factor of four to 50 MSPS. The horizontal scaling has also been expanded (200 ns/div - 10s/div) and the sample depth is now 1,000 points.

iMSO2 software improvements include the following enhancements:

- Waveforms can be paused and zoomed into.
- While zooming the horizontal and vertical scales, immediate feedback is now provided.
- Waveforms are sharper with more than four times the number of display points.

Both iMSO-104 and the new iMSO-204 work with free downloadable apps in the Apple App Store. Download the iMSO2 app with the iMSO-204 hardware and the iMSO app with the iMSO-104.

iMSO2 software version 1.1 is free in the Apple App Store. The iMSO2 app requires iOS version 5.0 or higher. Compatible products include: iPhone 5C*, iPhone 5S*, iPhone 5*, iPhone 4S, iPhone 4, iPhone 3GS, iPad Mini*, iPad Air*, iPad 4*, iPad 3, iPad 2, iPad, iPod touch [3rd (32 GB only), 4th and 5th* generation]. iMSO-204 hardware can be purchased for $399.97 from Oscium directly or from one of their partners.

*Devices require an official Lightning adapter to unlock compatibility.

For further information, please contact:

Oscium Website: **www.oscium.com**

Continued on page 63

Is your product innovative, less expensive, more functional, or just plain cool? If you have a new product that you would like us to run in our *New Products* section, please email a short description (300-500 words) and a photo of your product to: newproducts@servomagazine.com.

bots IN BRIEF

MONKEY BUSINESS

The interactions between animals and robots is always fascinating, and generally the more intelligent the animal, the more interesting the interaction. Researchers at the University of Portsmouth tried giving chimpanzees a robotic doll to see how they'd react, and the result was strikingly similar to humans.

Humans understand (mostly) that robots aren't alive, but that doesn't keep us from interacting with them in the same way that we'd interact with other living things. We look them in the eye, we talk to them, we respond to their movements, and we can even form emotional bonds with them.

Working with chimpanzees from Yerkes National Primate Research Center in Atlanta, researchers from the University of Portsmouth's Centre for Comparative and Evolutionary Psychology used a robot doll to test interactions with the animals. Called Robota, the doll was developed at EPFL's Learning Algorithms and Systems Laboratory, led by Prof. Aude Billard. It has a moving head and moving arms, and could make chimpanzee noises from a speaker in its chest. First, the chimps were shown humans interacting with the doll, and then they were given a chance to meet it themselves. Lead researcher Dr. Marina Davila-Ross describes what happened:

"Some of the chimps gave the robot toys and other objects, and demonstrated an active interest in communicating. This kind of behavior helps to promote social interactions and friendships. But there were notable differences in how the chimps behaved. Some chimps, for instance, seemed not interested in interacting with the robot and turned away as soon as they saw it."

The robot was able to imitate motions made by the chimps which they immediately recognized and responded to since imitation is an important part of social bonding. When the robot made more human-like movements, however, the chimps were significantly less interested. The reason that it's important to use a robot (instead of a human) in this research context is that you have complete control over the experiment, and through testing, it may eventually be possible to figure out what specific sounds and movements are used by chimps to (say) make friends with one another.

EVOLUTION CONTINUES

iStruct — the robotic ape from DFKI in Germany — can now stand up on its hind legs, making the transition from quadruped to biped that took humans like a million years to successfully pull off.

The aim of this project is to develop a robotic system as well as biologically inspired structural components which — if applied on the robotic system — effectively improve the locomotion and mobility characteristics. In order to achieve this goal, an improved perception of the environment and its own condition are needed. The structures are as self-contained as possible with regard to sensing, sensor preprocessing, control, and communication. The biologically inspired robot itself is an ideal test platform for foot and spine structures. These structures can extend the already existing locomotion behaviors of a robot and are used contemporaneous as carrier and sensor systems. This way, different functionalities are united in one construction unit.

bots IN BRIEF

SIGN OF THE TIMES

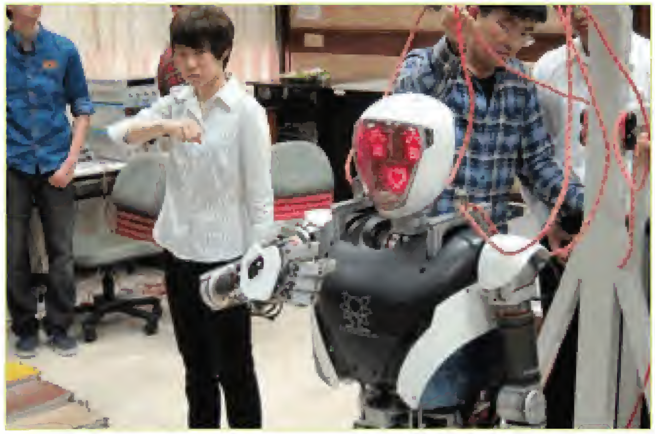
Unlike the DARPA robots, Nino — a humanoid unveiled earlier this year by the National Taiwan University's Robotics Laboratory — may not find itself performing tasks in dangerous situations any time soon, but this robot has some special skills. It is likely the first full-sized humanoid to demonstrate sign language.

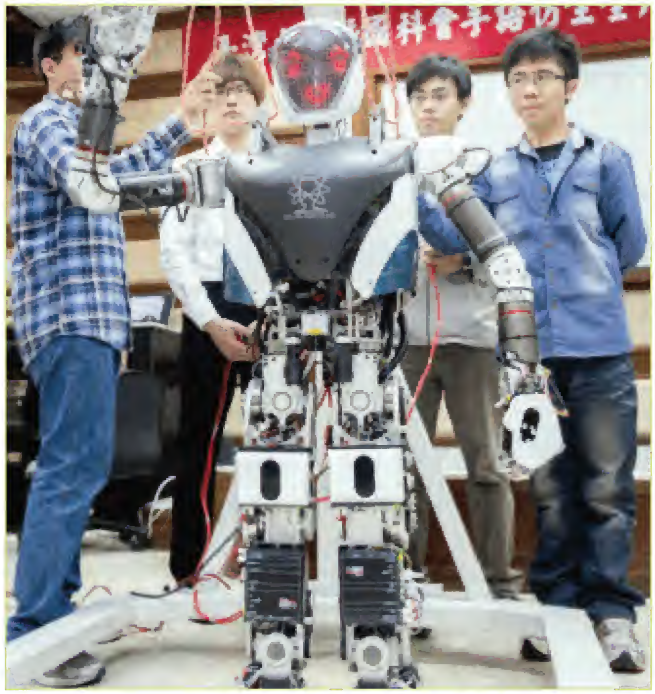
"Sign language has a high degree of difficulty, requiring the use of both arms, hands, and fingers, as well as facial expressions," commented Professor Han-Pang Huang, who leads NTU's Robotics Lab. That's why sign language, he added, is an ideal task to help researchers develop capable robot hands which could also find applications in factories or other kinds of work that require dexterous manipulation.

Nino stands 1.45 meters (4' 9") tall and weighs 68 kilograms (150 lbs). It has 52 degrees of freedom — including individual finger joints — and is equipped with 112 sensors that monitor the robot's motors, power usage, and temperature. An LED array in its head produces simple emoticon-style expressions such as blinking and lip movements. Nino is also able to walk, turn, and slowly climb up and down stairs and ramps.

One challenge for robot sign language is that mechanical hands with five fingers are usually considered slightly redundant for the type of work robots are expected to do. Advanced robots such as Boston Dynamics' Atlas or Willow Garage's PR2 feature only two or three fingers. Up until its latest revision, even Honda's ASIMO (considered one of the world's most advanced humanoids) didn't have hands with individually moving fingers. KAIST's Hubo robots are among the exceptions.

It took 20 researchers and students about three years to develop Nino, but they still have their work cut out for them. Although the robot was programmed to sign certain words in advance — such as introducing itself at a press conference — it doesn't have the image processing software necessary to interpret human signing which uses subtle movements and can be very quick. In the future, they'd like Nino to be able to hold a conversation with a person using only sign gestures.





TIME TO PLAY-i

These colorful robots are not only fun to play with, they can teach kids computer programming skills. That's what Play-i — a Silicon Valley startup founded by engineers from Google, Apple, and Symantec — says about its robots that were unveiled recently as part of a crowd-sourcing campaign.

The idea of using robots to teach kids programming, math concepts, and problem-solving is not new. In fact, it's been more than 40 years since MIT educator Seymour Papert demonstrated the possibilities of hands-on learning with his Logo programming language and mobile machines known as turtle robots.

Over the years, numerous robotic toys and kits designed for kids have come to market, but a lot of these products present a steep learning curve for kids (and parents!), and few work for very young children.

Play-i — based in Mountain View, CA — wants to see that change. It says its robots, Bo and Yana, can make programming fun and accessible for kids as young as five years old. The robots talk via Bluetooth LE with an iPad or other tablets, which Play-i says are the perfect interface for children to learn programming concepts in an engaging intuitive way.

Play-i is creating a tablet-based visual programming interface that lets kids build sequences of actions and learn through play and exploration. The sequences can start simple and get more complex as kids become more familiar with the robot and interface.

CLEAN SWEEP

Robots are all about taking things that humans don't like to do and doing them faster, smarter, and better. Or, in reality, doing them slower, dumber, and generally not quite as well, but doing them, nonetheless. What tends to be rare for robots is the capability to do something like a household task more efficiently than a human can. Moneual seems to have pulled it off with a new robot that can vacuum *and* mop at the same time.

The Moneual Rydis H67 includes a special "Shadow Active Cleaning Mode" which "detects and performs concentrated cleaning in areas that do not receive direct light such as underneath couches and beds, so your home is clean even in the places you can't see."

COMBAT ZONE

Featured This Month:

Post comments on this section and find any associated files and/or downloads at **www.servomagazine.com/index.php/magazine/article/december2013_CombatZone.**

SPECIAL REPORT:

Bot Builder's Buyer's Guide

● by Mike Jeffries

Welcome to the December Combat Zone!

Just in time for Christmas shopping, regular contributor Mike Jeffries has assembled a buyer's guide of three vital components for small sized bots. Wheels, motors, and speed controllers are available from a wide range of sources, and in an even wider variety of sizes, styles, and prices. (Prices shown are in US dollars unless noted otherwise.)

In addition to the exhaustive survey Mike offers, the pages of *SERVO* are chock full of suppliers for every part a bot builder could want — combat or those other kinds. Given, readers of this section have trouble understanding why you'd ever want to build a bot and NOT destroy it, but we understand that there are people who use machines for non-destructive fun.

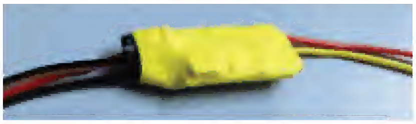
The Combat Zone is a resource for builders. We welcome your input, advice and, of course, articles about these or any other parts you've battle tested.

Product Spectrum:

ESCs for Bots 30 lbs and Under

When building a robot, there are a wide range of ESCs on the market. The primary vendors discussed here are The Robot Marketplace (**robotmarketplace.com**), FingerTech Robotics (**fingertechrobotics.com**), Equals Zero Designs (**e0designs.com**), and BotBitz (**botbitz.com**). The ESCs are being discussed from the perspective of their practicality in robots weighing 30 lbs or less.

BotBitz 10A ESC



ADD TO WISH LIST

Voltage: 3V–12V
Current: 10A peak
Dimensions: 0.91 x 0.71 x 0.32 inches
Weight: 9 g
Cost: $15 AUD
Comments: BotBitz approaches the ESC market from an unusual angle. Their motor controllers are actually reprogrammed Chinese brushless ESCs. The reprogramming is used to convert them to dual direction brushed motor controllers. Recycling these ready-made mass-produced motor controllers and reprogramming them puts BotBitz on the bottom of the price range while still having a solid reliable ESC.

With a 10A rating, the BotBitz 10A ESC is well suited for 1 and 3 lb fighting robots. If you're looking for advanced features, you'll have to look elsewhere for now. However, if you want a bare-bones ESC in a compact easy-to-use package, the BotBitz 10A ESC is a strong contender.

BotBitz 85A ESC

ADD TO WISH LIST

Voltage: 6V–24V
Current: 85A (1 minute)
Dimensions: 2.58 x 1.34 x 0.69 inches
Weight: 80 g
Cost: $75 AUD
Comments: This is the larger of the BotBitz reprogrammed ESCs. With an absolute 200A maximum and tests proving that it can handle 85A for a full minute without melting or catching fire, the BotBitz 10A ESC is a strong contender in the 12 and 30 lb classes if you're looking for a bare bones controller.

Holmes Hobbies BR-XL

ADD TO WISH LIST

Voltage: 6V–25V (2-6s lipo)
Current: 80A continuous
Dimensions: 1.97 x 1.54 x 0.83 inches
Weight: 45.6 g
Cost: $99.99
Features: BEC, regenerative brake, active hold brake, Castle Link programmable.
Comments: The BR-XL is the least robotics-specific ESC presented. These controllers were originally made for high powered RC rock crawlers. The features that were needed for these crawlers happened to closely match the needs of 12-30 lb fighting robots, so over the last few years they have seen widespread use. Unlike many RC car ESCs, the BR-XL actually appears to live up to the published specs, and has been used in the drive system on 30 lb fighting robots with high powered drive systems. These ESCs are durable, compact, and easy to use. For a 12-30 lb robot, they are certainly worth putting on the short list of options.

Holmes Hobbies has also recently released a BR Mini rated for 30A continuous and 80A peak with otherwise similar functionality and specs at a price of $69.99. They look promising, but I have not yet seen them in use in a robotic combat application.

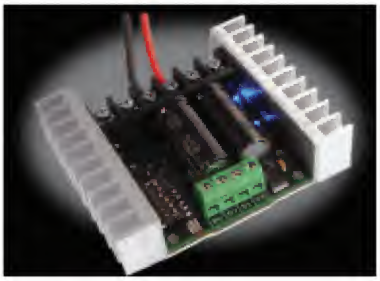
Equals Zero Designs Ragebridge

ADD TO WISH LIST

Voltage: 12V–36V, 50V absolute limit
Current: 30A continuous, 90A peak with limiting
Dimensions: 4.5 x 2.0 x 0.5 inches
Weight: 99 g
Cost: $185
Features: Adjustable current limiting, channel mixing, flip control, regenerative braking, BEC.
Comments: The Ragebridge is one of the newer ESCs on the market, and it has quickly gained popularity. The biggest feature on the Ragebridge is the adjustable current limit. This feature allows you to tweak the performance of the ESC to balance pure power vs. reliability. This also reduces stress on motors and batteries; when they're limited, you're never pulling the sometimes massive peak current loads, while still getting the vast majority of the usable power band of the motor.

Whether it's due to the way the regenerative braking code is written or just an aspect of how throttle ramping is handled, the Ragebridge also has an extremely smooth throttle response. For the 12 and 30 lb class, the Ragebridge is a fantastic motor controller option and should be considered in most applications.

Sabertooth 25 Dual Motor Controller

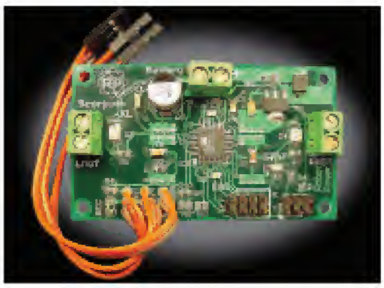


ADD TO WISH LIST

Voltage: 6V–30V
Current: 25A continuous, 50A peak
Dimensions: 2.6 x 3.2 x 0.8 inches
Weight: 96 g
Cost: $130.99
Features: Regenerative braking, current and overtemp protection, LiPo protection mode.
Comments: The Sabertooth 25 is a versatile controller, with many options for inputs and dip switches that allow a range of parameters to be quickly and easily changed. While versatile, the Sabertooth 25 can be fragile; a common recommendation is to coat it in a thick silicone adhesive (Shoe Goo and Goop are common options) and to shock mount it to reduce the risk of shock induced failure.

If you don't expect high shock loads in your application and you need the versatility the Sabertooth offers, it's worth considering.

Robot Power Scorpion XL



ADD TO WISH LIST

Voltage: 4.8V–28V
Current: 12.5A continuous, 45A peak
Dimensions: 2.7 x 1.6 x 0.5 inches
Weight: 28 g
Cost: $104.99
Features: Current and overtemp limiting, dual channel, flip control, channel mixing.
Comments: The Scorpion XL is the smallest dual channel ESC intended for 12 lb robots. With the low continuous current rating, this controller is best suited to weaponed robots with mid to low powered drive systems (if it is being used in the 12 lb class). The small size also means it is possible to put it into something as small as a 1 lb robot. However, trying to use it in anything below the 3 lb class is excessive. In the case of a 3 lb bot, it would be best suited to a rammer or wedge with a very demanding drive system. The small size and light weight make it a good candidate for 12 lb robots that are tight on both space and weight.

Robot Power Scorpion XXL

ADD TO WISH LIST

Voltage: 6.0V–28V
Current: 20A continuous, 45A peak
Dimensions: 3.27 x 2.27 x 0.5 inches
Weight: 52 g
Cost: $159.99
Features: Current and overtemp limiting, dual channel, flip control, channel mixing.
Comments: The Scorpion XXL is the larger cousin of the Scorpion XL. With the increased weight, size, and current limit, this controller would pair well with many 12 lb drive systems and may be capable of running lower powered or weapon focused 30 lb drive systems. The small size and light weight make it easy to fit into 12 and 30 lb robots that are tight on space and weight.

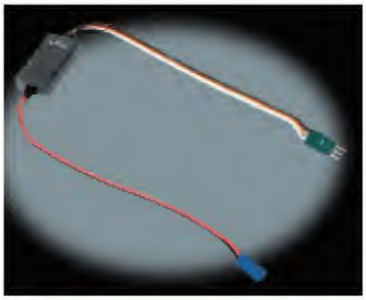
FingerTech Robotics TinyESC

ADD TO WISH LIST

Voltage: 6.5V–36V
Current: 2A continuous, 3A peak
Dimensions: 0.53 x 0.90 x 0.19 inches
Weight: 4.5 g
Cost: $33.99
Features: Undervoltage protection, overcurrent/temp protection, BEC.
Comments: The FingerTech TinyESC is — in effect — the default controller for the 150 g and 1 lb classes. Its small size, light weight, and wide voltage range make it a great option for most robots in those classes. The TinyESC has also seen use in the 3 lb class, though it hasn't gained the same grip here as it has in the 150 g and 1 lb classes. The TinyESC has been tested and is proven to work with some of the common drive motors used in 3 lb robots, though it is suggested that you solder the included 0.01 µF capacitor between the motor leads when using TinyESCs in a 3 lb bot. These ESCs are tiny, highly reliable, and offer smooth control.

If you're building an Insect class robot, these should be on the short list of ESC options.

VEX ESC (modified)



ADD TO WISH LIST

Voltage: 8.5V official, 14.8V reported
Current: 3A official
Dimensions: 0.6 x 1.1 x 0.21 inches without case
Cost: $9.99
Features: Modifiable.
Comments: The VEX ESC is an odd case in that you have to modify the controller to get the most out of it. When properly modified, the VEX ESC is a highly capable unit for the 1-3 lb classes. The modified VEX ESCs have been gaining in popularity, and recently have become available pre-modified through **TitanTechIndustries.com** for $19.99. If you're looking for a cheap, compact, no frills ESC for the 1-3 lb class, the modified VEX ESC is a solid option.

Special Note: Modification documentation is available at **http://thevariableconstant.blogspot.com/2013/05/thrifty-roboting-vex-motor-controller-29.html**.

Robot Power Wasp

ADD TO WISH LIST

Voltage: 6.5V-28V
Current: 10A continuous, 30A peak
Dimensions: 0.65 x 1.85 x 0.44 inches
Weight: 9.0 g
Cost: $69.99
Features: Full current limiting and overtemp limiting, limit switch inputs, BEC if used at or below 18V.
Comments: The Wasp is a compact single channel ESC from the same company that makes the Scorpion XL and XXL. The Wasp is best suited for use in 3-12 lb robots; however, due to its small size, caution must be used when attempting to use it in a 12 lb robot as many common 12 lb drive solutions can stall well in excess of the 30A limit.

While you may not fry the controller, the overtemp protection may result in the robot stopping in the middle of a match if you spend too much time near the current limit.

Product Spectrum:

Gearmotors for Bots 30 lbs and Under

There are quite a few good drive options available for robots 30 lbs and under. In the case of some of the gearmotors presented here, they are representative of several options and/or competing products that have essentially the same parts. They may be different ratios from the same manufacturer or several variants on the same gearbox from multiple sources.

30:1 Micro Gearmotor

ADD TO WISH LIST

(0-PL1093)
Dimensions: 0.94 x 0.47 x 0.93 inches (24 x 12 x 10 mm)
Nominal Voltage: 6V
RPM: 1,000
Stall Torque: 0.5 in-lbs
Stall Current: 1.6A
Shaft: 3 mm
Weight: 0.34 oz
Cost: $15.95
Comments: This gearmotor is one of many variations of the same design. Two of the major sources for variations of this gearmotor are **robotmarketplace.com** and **botbitz.com**. These gearboxes are frequently used in the 150 g and 1 lb classes, and are a decent option. In the 1 lb class, you will likely need four if you are not planning on a weapon focused machine. It is also advisable to protect or cover the exposed gears to prevent jamming or unnecessary damage during operation. The short shaft limits wheel options and makes external shaft support difficult. If you want an extremely light, compact gearmotor for a 150 g to 1 lb bot, this is worth considering.

FingerTech Robotics Silver Spark

ADD TO WISH LIST

(11:1 - 50:1)
Dimensions: 0.63 in (16 mm) diameter, 1.58-1.63 inches long
Nominal Voltage: 6V
RPM: 850 for 11:1; 189 for 50:1

Stall Torque: 0.35 in-lbs for 11:1; 1.6 in-lbs for 50:1
Stall Current: 1.3A
Shaft: 3 mm diameter, 1.5 inches long
Weight: 0.99-1.02 oz
Cost: $22.94
Comments: The FingerTech Robotics Silver Spark gearmotor is a very well made and durable drive motor option for the 1 lb class. It has been used in weapon focused 3 lb robots with a decent amount of success. With a wide range of ratios available (11.1, 22.2, 33.3, 50, 83.3, 100, 200, 300, and 600:1), it is easy to find an option that fits your needs. These motors are also capable of handling much higher voltages than their competitors, and have been run as high as 22.2V (though >10V does decrease motor lifespan) in a combat application.

The long shaft gives you plenty of options for both wheels and supplemental shaft support, though the latter isn't often needed in the 1 lb class. These gearmotors have relatively large pitch gears and because of that they're far less prone to stripping under load. This can often make the difference in a match and is one of the reasons they're very commonly found in 1 lb bots.

Kitbots 1,000 RPM Gearmotor

ADD TO WISH LIST

Dimensions: .98-.03 inches diameter, 2.14 inches long
Nominal Voltage: 12V
RPM: 930
Shaft: 4 mm diameter
Weight: 3 oz
Cost: $12-$24
Comments: There's very little in the way of dedicated gearmotors for the 3 lb class right now, and the Kitbots 1,000 RPM gearmotor is at the top of the pile. This somewhat generic looking gearmotor is effectively the off-the-shelf drive option for the 3 lb class. There are other gearmotors out there, but they're often either difficult to attain, out of production, or unrealistically expensive.

Kitbots offers a "battle hardening" service that improves the durability of these gearmotors for $12, and it's worth either having them do it or following the instructions they have posted and doing it yourself.

This motor is prone to voltage spikes, so it is recommended that with some compatible ESC options you pair it with a 0.01 µF capacitor to protect the motor controller.

Gimson Robotics GR02

ADD TO WISH LIST

Dimensions: 1.6 inches square (40 mm), 3.8 inches long
Nominal Voltage: 18V
RPM: 810 for 24:1; 542 for 36:1
Stall Torque: 106.8 in-lbs for 24:1; 160.2 in-lbs for 36:1
Stall Current: 61.6A
Shaft: 12 mm with flat
Weight: 1 lb
Cost: £29.80
Comments: The GR02 is a gearmotor from the UK and is meant to function as a replacement for modified drill gearboxes in the popular 30 lb class. With the size and weight of this gearmotor, it is also suitable for the 12 lb class. The shaft on the GR02 can be troublesome for some builders, however, they offer a wheel hub solution that pairs with the 12 mm shaft which minimizes this issue. They're also shipping the gearboxes with an external bearing kit which can be used to help handle the shock loads that often come with an overhung wheel drive setup.

BotBitz GM36 19:1 Gearmotor

ADD TO WISH LIST

Dimensions: 1.42 inches diameter (36 mm), 3.7 inches long
Nominal Voltage: 12V
RPM: 900
Stall Torque: 29.7 in-lbs
Stall Current: 52A
Shaft: 8 mm with flat
Weight: 0.93 lbs
Cost: $40 AUD
Comments: The GM36 is the Australian solution to the same problem the GR02 addresses in the UK. Cordless drill gearboxes are nice and off-brand models can be quite cheap, but year to year the design, motors, and ratios can all change, making it difficult to get a reliable known motor. These gearmotors are also meant for the 30 lb weight class, though are certainly compact and

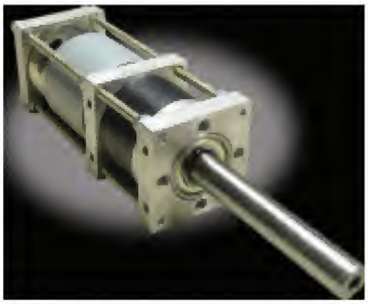
light enough for use in a 12 lb robot. The 8 mm shaft may be somewhat awkward to deal with; with that small of a shaft, it would be wise to either protect the wheel from shock or add an outer support bushing/bearing to distribute the load.

PDXNN Gearmotor/BaneBots P60 Gearbox

ADD TO WISH LIST

Dimensions: 1.5 inches square, 4.16 inches long
Nominal Voltage: 12V
RPM: 1,500 for 16:1; 900 for 26:1
Stall Torque: 91 in-lbs for 16:1; 145 in-lbs for 26:1
Stall Current: 148A
Shaft: 1/2 inch diameter with keyway, 1.5 inch long
Weight: 15.9-16.05 oz
Cost: $79.99-$84.99
Comments: Robot Marketplace sells an assembled version of the BaneBots P60 gearbox called the PDXNN, with NN being the ratio. Of the options, 16:1 and 26:1 are the most practical. BaneBots offers quite a few more ratios through their website. These gearmotors are a staple of the 12 lb class. They're not particularly expensive, are easy to use, and are reasonably durable. Three versions of the gearbox exist with each designed for a different common size brushed DC motor. If you take care to protect the shaft and wheels, these gearmotors are a solid option for the 12 lb class; if four or more are used, they're at least an option for the 30 lb class, though in that situation extra effort must be put into shaft protection. The addition of a shaft support is recommended.

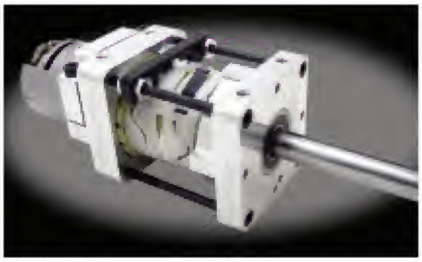
Robot Power Magnum 775



ADD TO WISH LIST

Dimensions: 1.75 inches square, 4.92 inches long
Nominal Voltage: 14.4V
RPM: 720
Stall Torque: 156.25 in-lbs
Stall Current: 140A
Shaft: 1/2 inch diameter with keyway, three inches long
Weight: 28 oz
Cost: $89.99
Comments: The Magnum 775 is the next step up from the P60. This gearbox is designed to work with an RS775 motor which is the largest option for the P60 gearbox. With a larger planetary gearbox and shaft supported by a dual bearing arrangement, this gearbox is a solid option for the 30 lb class. In a high shock environment, a motor support plate like the one shown in the photo is recommended; however, it may not be needed for all applications. This is the largest currently available gearmotor solution that could reasonably be wedged into a 12 lb robot.

Equals Zero Designs DeWut?!



ADD TO WISH LIST

Dimensions: 2.5 inches square, 6.1 inches long
Nominal Voltage: 18V
RPM: 450, 1,450, 2,000 selectable
Stall Torque: 162.4 in-lbs estimated, middle gearing
Stall Current: 155A
Shaft: 1/2 inches diameter with keyway, three inches long
Weight: 2.9 lbs with three inch shaft
Cost: $190 (assembled)
Comments: The DeWut?! from Equals Zero Designs is an attempt to create a replacement for the no longer available Team Delta power drive kit. Due to the size and weight, the DeWut?! is primarily an option for the 30 lb class. The three selectable gear ratios (51.2:1, 17.1:1, and 12.8:1) mean you have a solid range of speed options for most practical wheel sizes. These gearmotors are well designed, durable, and easy to use. Two DeWut?!s will provide plenty of power for a 30 lb robot, though if you want even more there have been competitive 30 lb robots made using four DeWut?!s.

Product Spectrum:

Wheels for Bots 30 lbs and Under

When building a robot, there is a wide range of wheel options to choose from. All of the wheels presented here are meant to be ready or nearly ready to go solutions that are practical to use on robots weighing 30 lbs or less.

BaneBots Large Wheels

ADD TO WISH LIST

Diameter: 2-7/8-4-7/8 inches
Tread Width: 0.8 inches
Weight: 62 g-143 g
Cost: $5.30-$6.80
Comments:
These wheels are often found on 12 and 30 lb robots. The range of available durometers (Shore A 30, 40, and 50), size options, low cost, and easy mounting make them a fantastic option for large bots. The 30A wheels have the best grip, but tend to wear out very quickly if you're trying to put a lot of power through them. The 40A wheels have a bit more longevity without sacrificing much traction.

The 50A wheels (which can be bought at **www.banebots.com**) are the hardest of the bunch and are durable enough that even aggressively driven robots can often go a full event on a set of wheels. The ease of use and near disposable price range make them one of the better 12-30 lb robot wheel options. The large BaneBots wheels are available with a range of mount options including two hex bores, two keyed bores, and two plain bores.

BaneBots Small Wheels With Hex Hub

ADD TO WISH LIST

Diameter: 1-3/8-2-7/8 inches
Tread Width: 0.4 inches
Weight: 11.4 g-28.4 g (wheel and hub)
Cost: $6.50-$7.69 (wheel and hub)
Comments:
The smaller BaneBots wheels are essentially the same as the larger wheels. The primary difference is that the small wheels are designed to use a separate aluminum hex hub. With bots weighing 3 lb or less, the 30A tread doesn't have nearly the issues with wear that it does on a 12 lb+ robot. You can use the higher durometer treads, but there's not much point.

These wheels give fantastic traction, but the design does have a drawback when compared to its competitors: The solid wheel and hub results in higher forces being transmitted to the shaft. If you use these wheels, make sure to support them well and — if possible — isolate them from the output shaft of the drive gearbox.

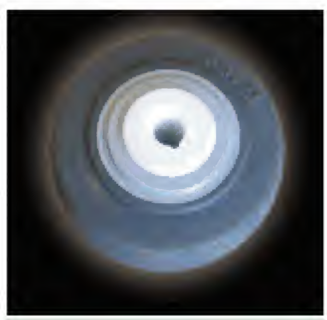
BattleKit Robot Wheel

ADD TO WISH LIST

Diameter: 4-10 inches
Tread Width: 1.5 inches
Weight: 1.1 lb-3.6 lb
Cost: $39-$49
Comments:
The Colson Performa is one of the longest used wheels in robot combat. It's a durable caster type wheel with a reasonable amount of grip, with a very long useful life. The BattleKits wheel provides the only ready-to-install chain driven wheel on the market. This wheel is designed to be mounted on a dead axle and driven by roller chain. This arrangement makes it easy to build a 4WD bot with only two drive gearboxes, or to build a drive system that protects the gearbox shaft from direct weapon impacts. (For example, the front wheels have the sprockets toward the robot body; the rear ones are away from the robot body with two sprockets on the gearbox shaft.)

One way to get extra life out of the wheel and hub combination is to buy extra casters and use a broach to cut the matching keyway in them. This can be done properly with an arbor press or (lacking that) a drill press and some creativity. Short of taking damage in combat, you can expect these wheels to last a long time.

NPC Drive Wheel



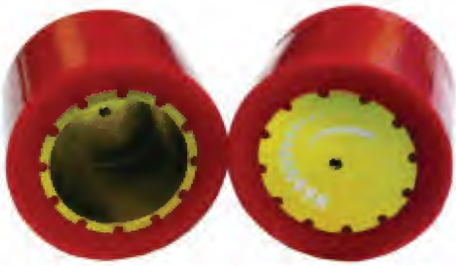
ADD TO WISH LIST

Diameter: 4-8 inches
Tread Width: 2 inches
Weight: 0.78 lb-2 lb
Cost: $24-$40
Comments:
The NPC drive wheel uses the same Colson Performa base with a

different attachment method. These wheels are designed to be used directly on a keyed half inch shaft.

If you want a durable wheel that can be run directly off of a gearbox, it's a very good option. The size and weight make it an option mainly for the 30 lb class, however, it would be possible to use it on a 12 lb robot.

FingerTech Sumo Wheel



Diameter: 1.75-3 inches
Tread Width: 1.5 inches
Weight: 61 g-178 g
Cost: $15.70-$31.32
Comments: As the name suggests, these wheels are primarily meant for Sumo competitions. They're available in two durometers (20A and 45A) and multiple bore options (3 mm-1/2 inch), and can have a keyway cut in them at larger bores. The smallest class these could be practically used for is the 3 lb class, however, they have been used in the 30 lb class in the past.

ADD TO WISH LIST

If you intend to use these for combat, a plastic insert could be added to the open side of the wheel to help dissipate the forces from any impacts the wheel sees to the shaft in a way that will reduce the chances of bending. While the 20A wheel option would provide fanastic grip, if you do choose to use these the 45A option will likely be much more cost-effective in the long run.

FingerTech Mini Sumo Wheel

Diameter: 1.125 inches
Tread Width: 0.85 inches
Weight: 12.6 g
Cost: $18.48-$20.48 (pair)
Comments: The FingerTech mini Sumo wheel was originally designed for their Mini Sumo kit; however, the high grip tread and lightweight construction make it an attractive option for 150 g-3 lb robots. If you're using the wheel on one of their Spark gearmotors or any other gearmotor of the same or smaller diameter, you have the option of having the wheel extend over a portion of the gearmotor. If the gearmotor is robust enough, this can allow you to build a much more compact drive system without sacrificing too much durability. However, this will likely block off the gearbox mounting points and may require creative mounting solutions.

ADD TO WISH LIST

Lite Flite/Lectra Lite With Lite Hub

Diameter: 1.5-5 inches
Tread Width: 0.445-1.5 inches
Weight: 1.2 g (hub only)
Cost: $4.62 (pair, hubs only)
Comments: This combination is one of (if not) the lightest wheel option for the 150 g-3 lb classes. With this arrangement, you take a stock Lite Flite or Lectra Lite wheel (except for the 1.5" Lite Flite, which needs to be bored to 5/32" from 1/8") and insert the hub, applying glue as it slides on. If the wheel is mounted externally, placing the hub on the outside will reduce the chances of the wheel being ripped off of the hub. However, this puts the shaft in greater danger of damage. The only downside with this hub solution is that you can't re-use it when the wheel is either worn or damaged beyond use. So, if your bot is prone to wheel damage, you'll want to keep a stash of spares available.

ADD TO WISH LIST

Lite Flite/Lectra Lite With Snap Hub

Diameter: 1.5-5 inches
Tread Width: 0.445-1.5 inches
Weight: 6 g-7 g (hub only)
Cost: $14.78 (pair, hubs only)
Comments: The FingerTech Snap hub is their more robust

ADD TO WISH LIST

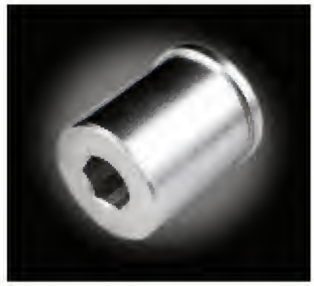
option for foam wheel mounting. These aluminum hubs use a main body with a slide-on plate to compress the foam wheel and a snap ring to hold it all together. These hubs allow for quick tread changes should the tread be worn or damaged, and are still very light weight. The design of these hubs also allows for a wide range of shaft sizes as the hub can be bored out to fit whatever size shaft your gearmotor of choice uses. This hub does put the shaft more at risk of damage since it doesn't absorb as much energy as the Lite Flite/Lite hub combo. However, the hub is small enough that it is possible to protect the hub from impacts without sacrificing too much weight.

Lite Flite With Dave's Hub

ADD TO WISH LIST

Diameter: 1.5-5 inches
Tread Width: 0.710-1.5
Weight: 9 g (hub only)
Cost: $20.99 (pair, hubs only)
Comments: Dave's Hub and the Snap hubs interface with the foam wheel in essentially the same manner, but the means of doing so on the Dave's Hub (compression plate screwed to hub) does limit the length of the shaft, making it impossible to support the shaft on the outside of the wheel. Some gearmotors are able to handle that sort of load, but if you do need to support your shaft you'll have to do it between the gearbox and the wheel.

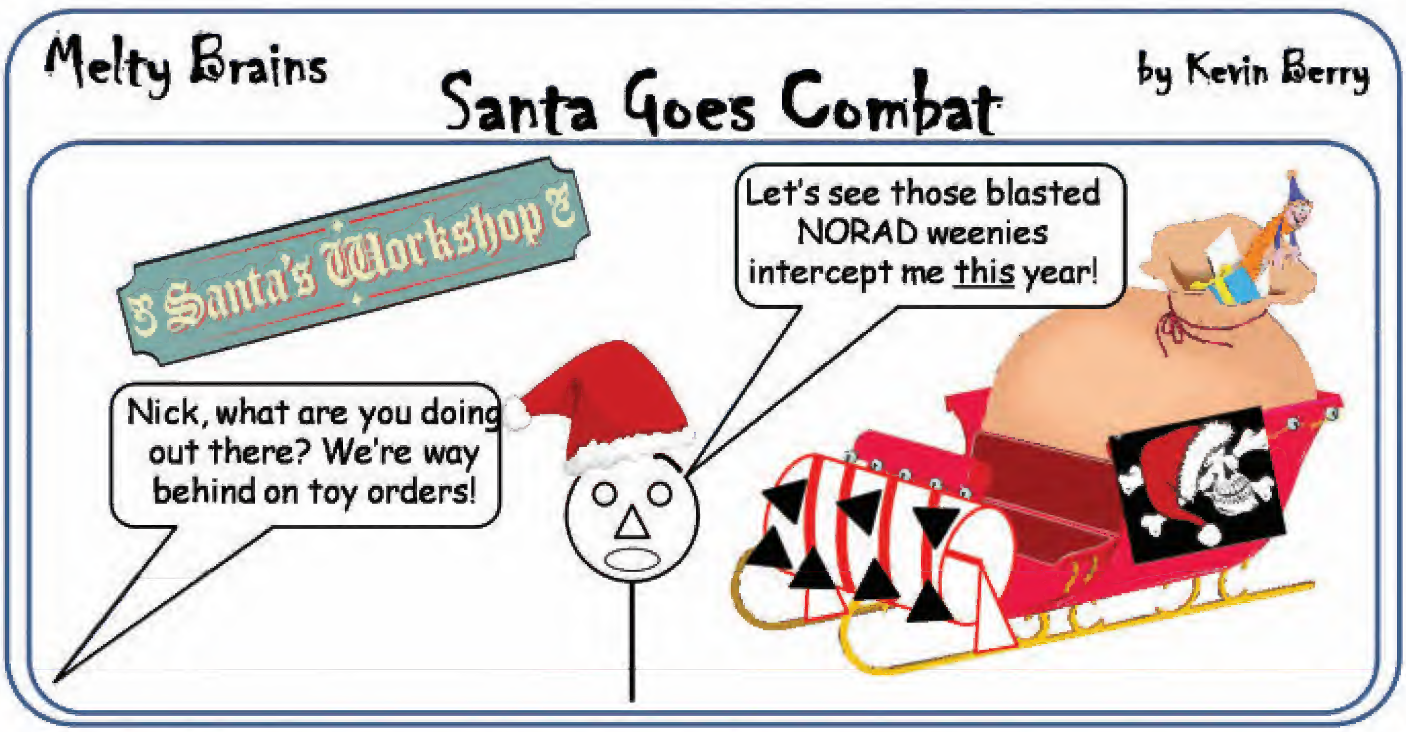
West Coast Products Live Axle Hub



ADD TO WISH LIST

Diameter: 3-6 inches
Tread Width: 1.5 inches
Weight: 0.15 lb (hub only)
Cost: $7.99
Comments: This live axle hub is designed specifically for a 1/2" hex shaft instead of the more common keyed round shaft. Like the NPC drive wheel, this hub uses a Colson Performa wheel that has had the hub pressed into it. If you decide that you want either a gearbox or driven shaft with a hex output, this is one of the only options. The design is simple and robust, so if your design calls for this form of mounting it should be a reliable and low cost option.

Hope this lists helps you get the bot parts you're really hoping for! **SV**



These wheels were discussed for their ease of use and suitability for use on robots weighing 30 lbs or less. If your application requires something that is not offered by the wheels covered here, you may be able to find something that will work at one of the following websites:

www.robotmarketplace.com/products/wheels_main.html

www.fingertechrobotics.com/products.php?cat=Wheels+%26+Hubs

www.andymark.com/Wheel-s/229.htm

www.vexrobotics.com/vexpro/wheels-and-hubs

http://teamwhyachi.com/wheels.htm

Building Builders

● by Matt Spurk

Nearly six years ago, my wife and I completed our greatest accomplishment. We had successfully made our first fully autonomous (sometimes too autonomous) biped walker. Since then we've completed a second walker, but now we're into the really challenging part of working with bipeds: raising them.

My two boys have been following me to combat robot events since before they could walk. My oldest son, Cayden, is turning six and my youngest son, Ethan is three and a half. Cayden has been to at least a half dozen events. Not too bad for someone whose hands are too small to hold the remote and drive at the same time. Ethan has been to probably three or four.

This article is a mini build report from my perspective of getting the bots and boys ready for their first event with their own "real" combots at Battle at Maker Faire, which took place on October 5th at the Orlando Mini Maker Faire.


Angry Bot under construction.

History

Cayden has previously competed in two events and Ethan one. They both fought in the unmodified toy class at the Gulf Coast Robot Sports event in Bradenton, FL. The GCRS unmodified toy class was exactly as it sounds: Intended for the youngsters, kids brought their small RC cars that were unmodified and tried to push each other into the pit. The two boys ultimately went 1-2 in a field of about five. Daddy was extremely shocked and proud.

I knew at that time that we had something that could really flourish. I just had to get them into something a little more competitive. That's when I took my old broken Antweight vertical spinner and bolted a wedge on the front. The robot Speedy was born and Cayden used it at another GCRS event.

The wedge spent a previous life as a VCR case. It was relatively lightweight, but was easy to work with and gave me a nice sharp wedge. Cayden did well driving it, but unfortunately facing the tougher bots with more driving experience was too much for him. The important thing about having his own bot and competing against "the big kids" was that he was bitten by the bug.

Building Angry Bot

I was in the garage digging through my spare robot parts trying to get inspiration for my next robot, when Cayden came out to join me. We have a very similar personality when it comes to robots. We look at

some random item that originally spent life performing some other task and immediately think it can be turned into a robot. In this case, Cayden grabbed out a galvanized steel dustpan that I had made in seventh grade shop class, and said "Daddy, I want to turn this into a robot!"


Practicing driving in the Mini Arena of Death.

Now, I had been thinking for years about converting this somewhat rusty old dustpan into a robot, and now I finally had enough inspiration. I decided this would make a good platform and even better it would allow me to teach the boys some shop skills. I grabbed out the spare electronics that I had: two Banebots 24:1 16 mm motors, two Banebots 3-9 ESCs, a GWS micro receiver, and a 7.4V 500 mAh Li-Po battery.

I was looking around the shop and noticed some PVC pipe lying in the corner. I decided to try slipping the Banebots gearmotor into the 1/2" PVC, and it was a light press fit. That's perfect for the drive mount. I went to Home Depot and picked up a pack of PVC clamps.

I now had a pretty solid motor mounting system (perhaps a bit heavy, but it was easy). I was able to hold the motors in the PVC with some hot glue and a little over the top to help support the gearbox. I also put the ESCs into the PVC and now I had a nice little drive package that could be installed in the robot.

Once I finished using the hot stuff, I let my sons drill the mounting holes for the PVC pipe clamps in the dustpan. We started by marking the hole locations. I held the pipe clamps in place while they took turns marking the pan with the permanent marker. We then moved over to the drill press.

We found safety glasses for everyone and then began drilling the holes. We started by drilling into a scrap piece of wood, so we could work on how fast to pull the handle, and then moved on to the bot. I aligned the holes and made certain they kept digits clear of the spinning sharp bits. It went very well. All the holes were in the right-ish area and everyone had fun (they wanted to drill everything).

Unfortunately, Dad is a worry-wart and his heart can only take four holes each, so we called it a night.

The following day, we began mounting up the drive motors. Both boys can run a screwdriver, so when it comes to turning screws I usually just let them go to work. We got everything bolted up and now the pressure was really on me. My sons were eager to start battling, but Angry Bot (who at this time was nameless) needed to be wired up.

I went back out to the garage after they went to bed and started soldering up all the electronics. I got everything finished that night, so the next day was going to be our first robot battle. Unfortunately, the robot had enough grip and enough torque that when you applied full throttle the bot immediately flipped on its back. It was a simple fix. I taped a piece of bent perforated aluminum under the robot that extended out the back and prevented the robot from flipping. We were ready for robot fights.

Building Scary

Scary was Cayden's old bot, and like younger brothers everywhere he gets the hand-me-down. Ethan also wasn't very interested in robots when we started building. I wanted to give him a faster fun robot, but I also wanted him to practice driving as soon as possible because that is the fun part for the youngsters. To get Scary ready to fight, I installed some new electronics and put in one of my old FM radios from my larger bots because it had a programmable remote. After a short while of instruction manual reading, he had a robot that had the speed dialed down to controllable and he was ready to fight his big brother.


Both bots after paint.

Practicing

I know a good driver with an

okay robot can beat a great robot with a bad driver, so we needed some practice. We began by simply drawing a big square with sidewalk chalk on the driveway. The last bot remaining in the square was the winner. We made a big square on the order of 12' x 12', so their random stick mashing style didn't result in two bots immediately driving out of the square.

The problem was they still had issues driving and they kept dropping the remotes. I watched their hands closely and noticed their little fingers were too short to hold the remote and move the joysticks. I pulled out some chairs and voilá, the driving improved immediately.

The boys started getting even better with each match. The first couple matches only lasted a few seconds before someone drove out (without ever touching each other). After a few days, they started being able to run into each other occasionally before immediately driving out of the square. Then, the bad stuff started happening. I was letting one of Cayden's friends try driving Angry bot when suddenly one of the wheels fell off. I assumed the screws just vibrated loose.

On further inspection, it turns out the gear case itself had failed allowing the face mounting bolts to fall out. I realized my mistake. I had never put an outboard support on the BaneBot gearmotor.

I knew that was a big no-no, so I took Cayden's robot back in and began building an outboard support for it. It took me a couple days, but we got it back up and running.

The new setup was much stronger. I also used the shaft support as a means to keep the robot from tipping, so I was able to remove the bent aluminum plate. The boys were back in business fighting in the driveway.

At this time. I started to notice that Cayden was getting some pretty good wear on his wheels, so I decided it was time to break out the plywood and build a ... wait for it ... arena. It's getting serious now. By arena, I mean a 4' x 4' plywood floor with some 1 x 4s nailed to the edge to provide some walls and an opening on one side. We broke out the house paints and had some fun decorating up the arena with shapes and pictures.

After a few days of practicing in the new arena, we decided these robots needed some paint too, so we went out in the garage and picked out our favorite colors. Cayden went with red and Ethan went with green. Both boys wanted to try to paint them, and I couldn't come up with a reason why not to, so we went for it. With my help to push the button down, they actually did pretty well. After a couple coats of color, the boys explained that they wanted their robots to look like things. Cayden wanted his robot to look like an Angry Bird and Ethan said he wanted his to look like a scary monster.

Ethan explained that he wanted a scary robot, so the other robots would just run away and fall in the pit. (That's a pretty good strategy for a wedge bot!) I certainly couldn't argue with his logic. I broke out the pencil and permanent marker, and after an hour or two (like eight) I finally had an ogre and an angry bird. Once they had paint, the boys decided to name their robots Scary for the scary monster and Angry Bot for the Angry Bird.

I applied a couple coats of clear-coat to help protect their "masterpieces," then we got the bots back together again for a couple more days of fighting. We had a week to practice, so all should be well. We started honing our skills and after a few fights, I noticed Cayden's robot started to pull to the left. I picked up the bot and one drive motor face was loose. I tightened the screws, but one didn't feel like it really tightened. I decided to open the case and check it out, and my worst fear was realized. The gear case had shattered again, and I didn't have any more replacements. I decided to order a pair of the 20:1 FingerTech Spark motors from Robot Marketplace. I went with the gold series, and I was not disappointed.


Ready for the big day!

The motors arrived on Wednesday and on Friday we had to leave for Florida. I took out the old PVC mounts and made a one piece mount for both Spark motors. This saved me two steel PVC clamps, so I had some more weight to spare. I got everything wired up and reweighed the bot. With the weight savings, I was able to add rear armor. Boy, did that turn out to be handy. Cayden and Ethan had one night to practice driving for about 20 minutes and then it was bed-time. We were ready to head to the Orlando Mini-Maker Faire.

Next month, I'll write about how the event went, and the answer might surprise you. **SV**

Continued from page 6

speed best when the environmental conditions are relatively constant. Like the purely proportional controller, the predictions made by a PI controller are predisposed to instability with some variable combinations. To counteract this instability, a derivative term (D) — the equivalent of a smoothing capacitor in a power supply — can be used to suppress the instability. Tuning a PID controller consists of optimizing the values of the three constants: proportion constant, integral constant, and the derivative constant.

If you've worked with a PID controller, you've no doubt discovered that there's no guarantee that it can predict the future speed or velocity of your robot platform. Not only do you have to be smart about adjusting the constants, but the conditions must be repeatable. If you tune a PID controller for the hardwood floor in your living room, don't expect your robot to operate smoothly on your lawn.

Of course, the PID controller isn't the final word in predicting, say, velocity. There are numerous variations and enhancements to the basic PID algorithms, as well as more sophisticated algorithms such as various flavors of the Kalman filter. The Kalman filter has a number of potential advantages over a PID controller, including relative resistance to noise. The downside is that the computational overhead favors a Parallax Prop or Raspberry Pi over a simple Arduino or PIC.

Given the availability of low cost, high performance microcontrollers, I predict that we'll be seeing robots with more sophisticated prediction algorithms — whether for tracking movement with a servo-based camera, the speed of a drive motor, or the relative RPM of quadcopter rotors. **SV**

GET YOUR ROBOTICS ON!

By L. Paul Verhage

Post comments on this article at **www.servomagazine.com/index.php/magazine/article/december2013_Verhage.**

One of the goals at the Boise Robotics Group (The BoRG) is to teach folks how to make robots. The BoRG's latest effort tapped into the growing do-it-yourself movement. We offered to teach eight families how to makes robots from the ground up through the Boise School District's Community Education. This article describes what we taught participants and some of the lessons we learned ourselves.

BACKGROUND

The BoRG has endeavored from its inception to bring robotics to the general public. If successful, our hope was membership would grow as a result. We find however — as I imagine most robot clubs do — that more people are more willing to begin a single robotics project than to build robots as a long-term hobby. This means robot clubs must do a lot of fishing in an effort to hook that one person or family. The BoRG's latest attempt to bring robots to the people and the people to our robotics club was through our local community education.

Community education is a program — often offered through a public school or university — to provide informal educational opportunities to the public. People with expertise in subjects like say, starting a business or practicing yoga, arrange to teach these subjects. Community Education then handles student registration and assigns a school or university classroom where the instructor will teach the course for a short period of time like one month, for example.

Late last year, the BoRG started developing a Community Education class focused on building your first robot. The school district's Community Education program both encouraged the class and helped us refine it. That's why at the end of last April, we found ourselves teaching a beginning robotics class to eight families.

THE ROBOTICS CLASS

Over our four week course, families constructed the brain and body of their first robot. That was a tall order for most since they had never soldered or programmed before attending this class. We suspected this would indeed be the case, so we developed a kit with all the tools needed to make a robot. However, the kit wasn't enough. With this many beginners in one classroom, participants class needed the help of mentors if it was going to succeed. The mentors were volunteers with soldering and programming experience. They were members of the local chapter of Mensa, the BoRG, and the Voice of Idaho amateur radio club.

The box of goodies that each family taking the class received. Believe it or not, there's a robot buried in this box just waiting to get out.

THE ROBOT KIT USED IN THE CLASS

One of the problems we faced was that robot kits can be expensive enough to turn off perspective students. Since we designed the class for beginners who may never take up robotics again, the robot kit required for the class needed to include all parts and tools required to assemble the robot (except for the programming laptop and batteries). It had to remain inexpensive enough that it didn't turn people off.

We met our goal by designing a kit around the CheapBot-14 robot. The kit with all its parts and tools came out to $109.37 per kit. We kept the cost low by purchasing inexpensive (but more than adequate) tools and finding sponsors who donated funds to offset the price. The total cost of class registration was $70. Half of that was for community education's costs and the other half was for the kit. By allowing two family members to register for the class, we (in a sense) cut the cost for the class in half.

ROBOTICS CLASS ACTIVITIES

We began the class by teaching our students how to solder. The assumption was that if we started the students soldering rather than started them listening to lectures, they would be more motivated to complete the class. Participants did watch a small Power Point presentation, but this was as they were plugging in their soldering irons and bending the leads of resistors.

Next, students opened their robot controller kit and identified the components inside as we displayed them in a second Power Point show. After they were acquainted with soldering and identifying components, they began soldering their robot controller. We outlined each step of its construction in a third Power Point tutorial.

After completing the robot controller, we instructed students to test their creation. This way, they would identify potential errors prior to installing the PICAXE-14M2 microcontroller. We guided students though the testing of

A student and his mother waiting for the soldering to get hot enough to begin soldering.

SKILLS TAUGHT

We determined that our families needed eight skills to complete a robot. There wasn't enough time for them to become experts in all eight skills, considering the limited amount of class time available. Our goal, therefore, became to make them somewhat familiar with the following eight skills.

- Soldering components to a printed circuit board (PCB).
- Using a digital multimeter to test electronic circuits.
- Identifying popular electronic components by sight.
- Interpreting popular electronic schematics.
- Reading resistor and capacitor values.
- Mechanically assembling a robot body.
- Programming a robot to reach a goal using BASIC.
- Incorporating feedback in a robot program to guide a robot's behavior.

their robot controller with another Power Point show, which illustrated the following four tests:

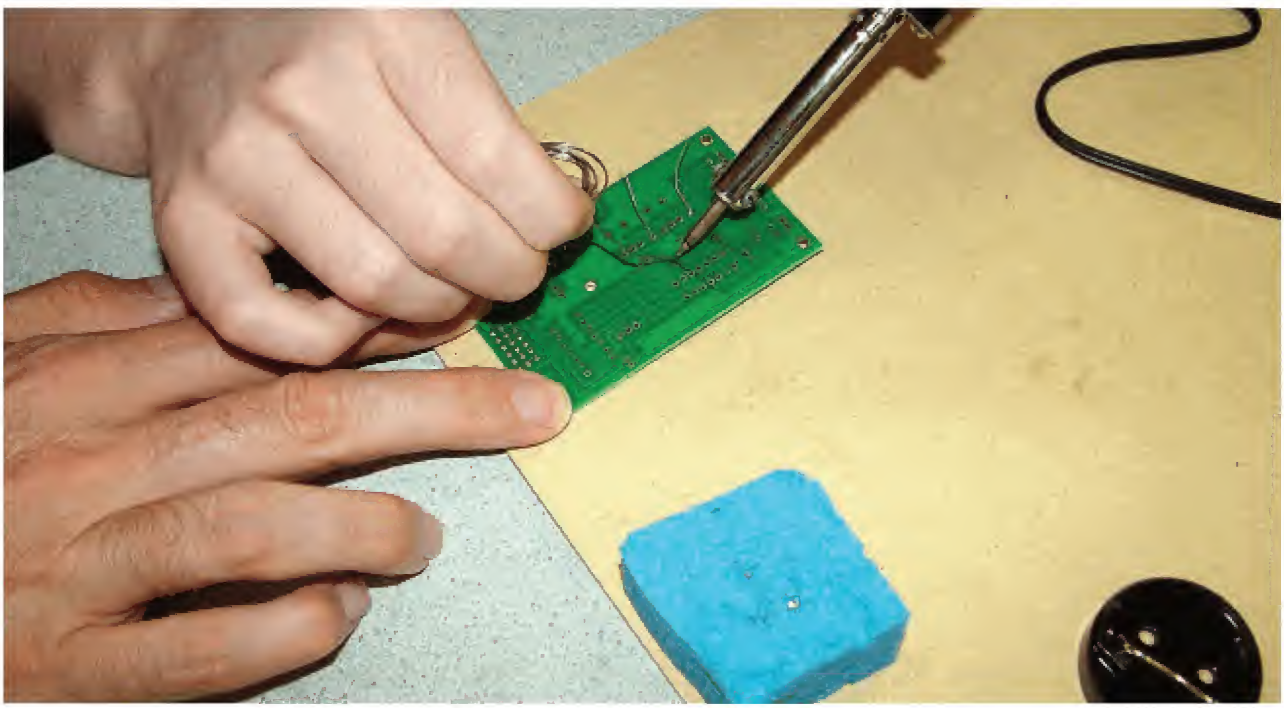
1. Make sure all the solders looked good. This meant solders formed shiny cones that did not overflow their pads.
2. Use the DMM to verify there was no continuity between the power and ground terminals of the battery holders.
3. Use the DMM to check that proper voltage existed between the power and ground pins of the PICAXE IC socket. The proper voltage was stated as being between 4.75 and 5.25 volts.
4. After inserting the PICAXE-08M2, program it with a simple debug program to test that two-way communication existed between the PICAXE and laptop.

Before assembling the robot body, we gave students two programming assignments. First, they programmed their robot controller to increment a variable. This showed them how to program their robot controller count to 255. The second assignment was to make an LED blink. They then adjusted the length of the *PAUSE* command and observed its affect on the LED's rate of blinking.

MAKING A ROBOT BODY AND NAVIGATING USING DEAD RECKONING

After creating their first programs, students began building their robot body. Again, we lead them through the construction with a Power Point presentation. After constructing the robot body, students mounted their robot controller to it. To drive the robot, we showed class members how *HIGH* and *LOW* commands controlled the spinning of the robot's wheels. Since the robot had two driven wheels and tail dragger, students learned to make their robot travel forward, backwards, turn left, turn right, and stop by controlling which direction each wheel turned. We did not teach our students to use pulse width modulation as a means to control the speed of each wheel.

After demonstrating that they could program their



A student's first solder. One of the mentors is stabilizing his PCB.

COMPONENTS PROVIDED IN CLASS

The robot kit was pretty extensive as the following list shows:

- CheapBot-14 robot controller kit (from NearSys)
- CheapBot robot body kit (from NearSys)
- CheapBot line follower kit (from NearSys)
- Plastic parts cup (plastic jar lid from a container company)
- Wire cutters (jewelry cutters from Walmart)
- Soldering sponge (cut up sponge from Walmart)
- Cen-Tech digital multimeter (from Harbor Freight)
- Screwdrivers (from Harbor Freight)
- Safety glasses (from Harbor Freight)
- Pliers (from Harbor Freight)
- Boxer 30W soldering iron (from **Frys.com**)
- Work surface (Masonite sheet form Home Depot)
- USB-to-serial adapter (from Parallax)
- 1 GB USB thumb drive (from **Amazon.com**)*
- Soldered LED and resistor (scraps from a junk box)
- Copy of *SERVO Magazine* (Thanks T&L for your gift)

* Contains PICAXE editing software, lessons, and software driver.

robot to travel, we explained the benefit of using subroutines. Students learned how subroutines shorten the length of their programs while making it much easier to understand.

Now that they had robots that could drive, they could control the direction each wheel spun and could understand the effects of the PAUSE command. So, we gave students the task of programming their robot to drive in a square path. We explained that this is called dead reckoning and that sailing ships used this method for navigation in the 15th century.

Dead reckoning is a method for driving a robot that does not make use of feedback. As a result, robots using this technique eventually deviate from their programmed path. The reason is that the wheels do not turn at exactly the same speed. There are ways to address this issue; in fact, one of our families corrected for this behavior by occasionally halting the faster spinning wheel.

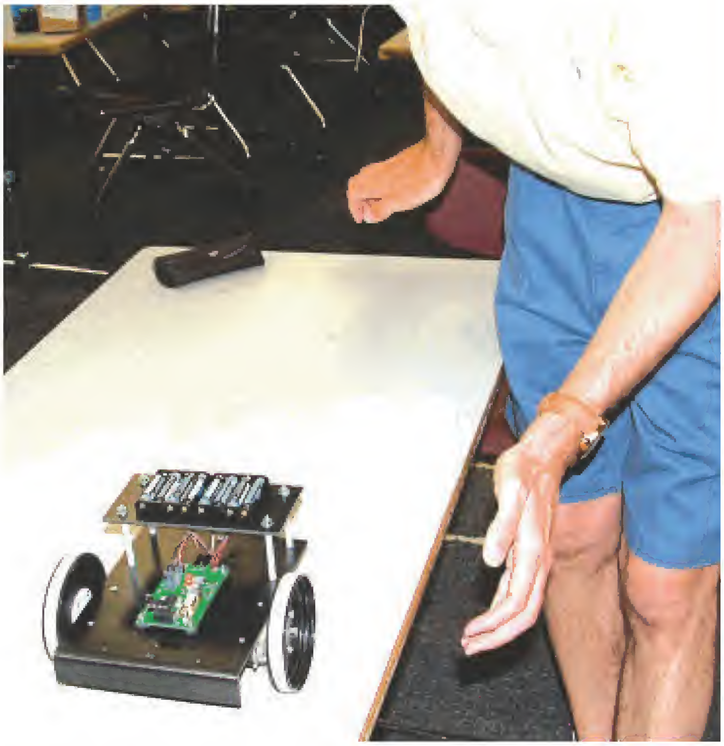
James, a mentor from the BoRG is helping a student program his recently completed robot controller.

USING FEEDBACK FOR LINE FOLLOWING

We asked students to imagine trying to walk to the door with their eyes closed. Even if they aimed themselves accurately towards the door before closing their eyes, each step was likely to take them off course. This mental exercise and observing the behavior of their robot made our students aware of the need for feedback in their program.

Because this was a beginner's class, the method of feedback we chose was line following. Students found themselves soldering a second but simpler PCB. The line follower consisted of a left side and right side detector. Each detector consisted of a phototransistor and neighboring IR LED. If the light from the IR LED reflected off the surface below it, then there was no black tape beneath the detector, so it produced a signal of zero volts (a logic low). The opposite was true when the black tape was beneath the sensor.

After completing their line follower circuit, students bolted it to the front of their robot and plugged its cable into the robot controller's GPIO. Then, we explained how to use the *IF-THEN* command to control the driving of the robot's wheels. At this point, we were running out of class time. Several of our students needed extra time to catch up with the rest of the class, so they began programming their robot at home.



One of our adult students testing whether or not his robot drives in a straight line.

We plan to use what we learned to change the class for 2014. These improvements are expected to increase our success next year. I also believe these changes will put the Boise Robotics Group into a good position to teach beginning robotics to larger classes in the future. **SV**

The *SERVO Magazine* included in each kit was a hit with our students.

LESSONS LEARNED

At the conclusion of the class, everyone had a working robot. They also had the tools and experience to make additional robots. Everyone could program his/her robot to drive around using dead reckoning. There were a couple of folks who needed extra time to complete their line follower challenge. As a result, we learned a few things about teaching this class to keep in mind for next time:

1. Place a checklist in each kit to show we added each component to the kit. We had a few robot kits that were missing parts.
2. Keep the microcontrollers out of the kits and instead, hand them out after students complete their robot controllers. We observed several instances of people inserting the PICAXE into its socket prior to soldering it, and we also had to replace a few PICAXEs that got lost.
3. Have mentors tin the soldering irons for their students when they plug them into the outlet. Some of the soldering iron tips became oxidized when students didn't stay on top of the tinning.
4. Include sample programs on the thumb drives so that families falling behind can try them out at home. This will help students develop their programs and catch up with the class if they fall behind.
5. Add a better thermal barrier around each ground pin so they are easier to solder. Pins fully connected to ground planes require more heat from the soldering iron. This need for additional heating creates a risk of damaging the PCB.
6. Make the solder pads a little larger. Some solder pads for thin diameter leads were a bit too tiny to solder without risking the solder spreading to neighboring pins.
7. Develop code and procedures that test assemblies prior to bolting them together. It's a major pain to remove and fix defective PCBs from the robot. It's better that students test and correct their circuit when it is easier to orient and fix them.
8. Introduce the class to the documentation stored on the thumb drive. This way, they will know what's available and where to find it. This information is helpful to students who fall behind.
9. Create a basic program to run the robots and line followers. The program should let students test their robot, but requires them to improve it if they plan to use it.

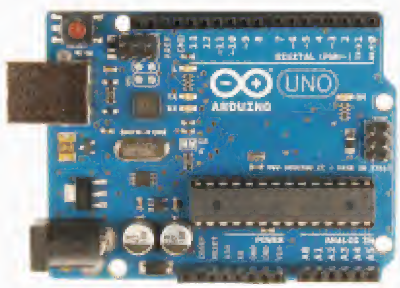
CanaKit™

Electronic Kits • Modules • Parts

**Save 10% using Coupon Code SM2012*
when ordering online at www.canakit.com**

Motor Controllers • Arduino • Arduino Kits • Arduino Shields • Robot Platforms • Parts

Arduino Uno



$29.95

ARDUINO-UNO

Latest "R3" Version!

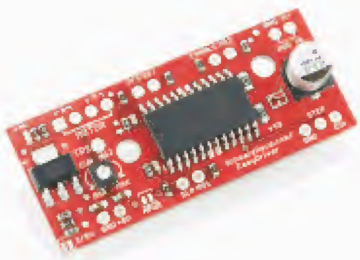
Inventor's Kit for Arduino

ARDUINO-PRK

$94.95

Includes 36-Page Printed Full-Color Manual

EasyDriver Stepper Motor Driver



SX09402

$14.95

Stepper Motor

SX09238

$19.95

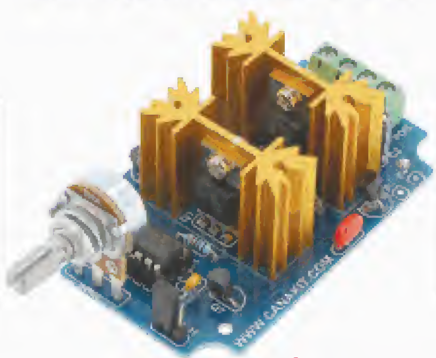
50A Motor Speed Controller (PWM)

UK1133

Optional LCD

$59.95

7A Bi-Directional Motor Speed Controller (PWM)



UK1125

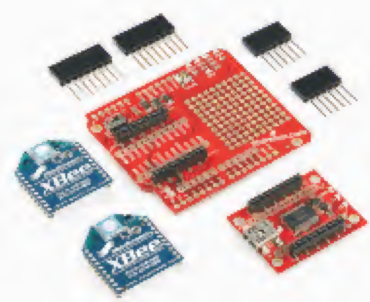
$39.95

USB PIC Programmer (MPLAB Compatible)

UK1300

$69.95

Xbee Wireless Kit



XBEE-KIT

$94.95

1•888•540-KITS (5487) info@canakit.com

www.canakit.com

* Limit one per customer. Can not be combined with other coupons or volume discounts. Does not apply to products already on sale. Prices subject to change without notice.

Disability in the Modern Age: *How Rehabilitation Robotics is Changing Lives Across the World*

By Morgan Berry

Post comments on this article at **www.servomagazine.com/index.php/magazine/article/december2013_Berry.**

A few months ago, I was in a very severe car accident. After a head-on collision, both my arms and both my legs were broken in multiple spots, requiring a massive surgery on three of my four limbs in order to repair the damage with metal rods and plates. Not surprisingly, I needed a great deal of physical and occupational therapy to relearn how to do everything from walking to holding a spoon to tying my shoes, and countless other things I had taken for granted before the accident. One thing that struck me through this process was the creative ways my occupational therapists adapted simple items to aid in my therapy.

Tiny beads hidden in Silly Putty were used to help improve the fine motor skills in my fingers. They taught me to use a trigger grip reaching device to put on my socks. In fact, for every problem I came across, my therapist would dash from the room and return with an almost laughably simple device that worked exactly the way I needed it to. Of course, there were plenty of high-tech and complex items used as part of my therapy too. For example, in an attempt to quicken the regeneration of nerves that were damaged, my physical therapists employed a machine that sent electrical impulses into my arm.

It is not surprising that robotics devices are finding their way into the field of physical therapy. Researchers are currently working to create more functional prosthetic limbs, exoskeletons to allow patients with paralysis to become mobile again, and tools used for therapeutic purposes. While the disabled were once limited to clunky and uncomfortable devices, now these patients are being given options that help to make use of what functionality their bodies still have.

More and more, therapists are turning to the burgeoning field of rehabilitation robotics to aid in the treatment of victims of short term injuries — like me — in addition to individuals with permanent disabilities. Robotic devices have the potential to revolutionize the way patients are treated and change the lives of disabled people across the world.

Rehabilitation robotics actually began with research into exoskeleton devices back in the 1960s. An exoskeleton is essentially a wearable robot, with a system of motors or hydraulics designed to assist the wearer by increasing strength and endurance. Of course, this idea has been conceptualized in fiction since the debut of Stan Lee's Iron Man comics in 1963, but exoskeletons have been in development in the real world for decades. In the United States, the goal was to use these exoskeletons as a way to enhance able-bodied soldier's combat abilities.

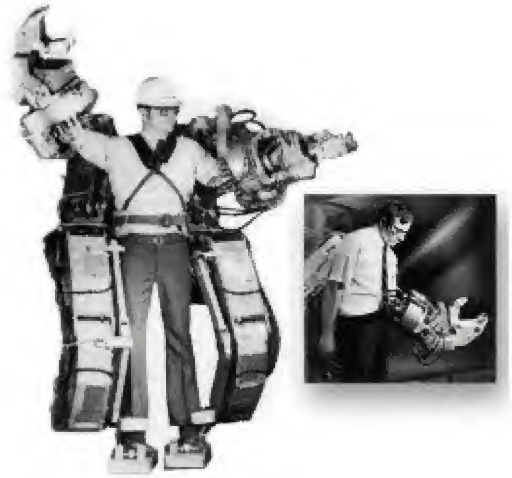
The Hardiman — which stands for Human Augmentation Research and Development Investigation — was developed by General Electric in 1965. The goal was for Hardiman to allow the user to lift 1,500 lb, but the exoskeleton was severely limited by the technology of the time. So instead, researchers went about making just the arm of the Hardiman which they hoped would be able to lift 750 lb.

Although it could successfully lift this amount of weight, the arm weighed three quarters of a ton — double the amount it could lift — making the device highly impractical in a real world application.

In the 1980s, workers at the Los Alamos National Laboratory developed a suit designed to enhance strength that responded to the wearer's brain activity through sensors in a helmet. Again, the limited technology of the time presented problems for the machine's success. The computers were too slow to allow the device to function in a smooth and succinct way. It was also too bulky to be portable.

Although these early attempts at developing an exoskeleton were not entirely successful, they provided a foundation for future projects which were quick to emerge as technology began to improve. So far, exoskeletons have not yet reached wide use, but they have the potential to revolutionize life for people with limited use of their limbs.

One manifestation of this is the Ekso bionic suit. The Ekso was developed to allow formerly wheelchair bound paraplegics to walk. The device contains a computer that



The Hardiman exoskeleton — while ambitious — was severely limited by technology of the time. *Photo courtesy of General Electric.*

The Ekso exoskeleton allows paraplegics to walk again, some of whom have been confined to a wheelchair for decades. *Photo courtesy of Ekso Bionics.*

This 3,000 year old prosthetic toe is the oldest example of a prosthetic. It was found attached to a mummy who may have lost the toe as a result of complications from diabetes. *Photo courtesy of the BBC.*

can sense when the user is ready to walk; it relies on the wearer to shift the weight in his/her hips before beginning the next step. The Ekso contains four motors — two at the knees and two at the hips — to power the walking motion. The controls can be set to provide more or less assistance to the patient, so the user can actually exert force from the body to move the device. Because of this, the patient builds muscle and works to alleviate the physical ailments that can come with life in wheelchair.

Currently, the device is commercially available for hospitals; patients can utilize the exoskeleton under medical supervision during physical therapy sessions. In 2014, the company plans to release a personal version meant to be worn by a patient on a regular basis.

In the past, prosthetic limbs were primarily worn for aesthetic purposes, not functionality. For thousands of years, prosthetics were made of wood or metal, and attached to the body with leather straps; the earliest example of this is a 3,000 year old wooden toe discovered in Egypt. Knights in the Middle Ages wore artificial limbs made of iron. Heavy and not very functional, these were worn to conceal the fact they were missing limbs, which was considered an embarrassment at the time.

The metal limbs worn by knights in the Middle Ages were most likely built by the same metalworker who constructed their armor. These prosthetics were primarily worn to conceal the missing limb.
Photo courtesy of the BBC.

In the 1500s, French surgeon Ambroise Paré invented a prosthetic leg that featured a locking knee as well as a hinged mechanical hand, kicking off the trend of improvements over the traditional prosthetic devices.

As medicine and technology has improved, so have prosthetic limbs, with technological innovations such as the development of plastic impacting the design of the devices. As medical knowledge improved, more patients ended up surviving amputation surgeries, increasing the demand for more functional prosthetics.

It is no surprise then, that robotics is also revolutionizing this area of rehabilitation. Scientists now have the ability to use signals from the brain to control prosthetic limbs. In some instances, this is done through electrodes placed on the skin to sense electrical signals in the remaining nerves. In other cases, the remaining nerves are directed to control an unaffected muscle through a surgical procedure.

For example, the nerves that once controlled a patient's arm would now control part of a muscle in the chest. The patient can then learn to contract these muscles in order to signal the robotic prosthetic to move. Currently, bebionic and i-limb are advanced hand prosthetic options for amputees. These products allow individualized movement of each finger, allowing patients to type and eat with the hand.

The i-limb also features a grip control system that locks the fingers into position when gripping an object after detecting that the proper amount of pressure has been applied. This allows the user to confidently grasp an item — like a soda can — without worrying that the hand will apply too much pressure and crush it.

The next step for robotic prosthetics is likely the incorporation of a brain-computer interface. A brain-computer interface is a system that measures electrical impulses in the brain and then feeds this information to a computer where it can be used in a multitude of ways. Scientists are using these interfaces to create prosthetics that are even more realistic than their already impressive predecessors.

By directly implanting electrodes onto the brain, scientists at the University of Pittsburgh and Brown University have successfully allowed subjects to use thought to control a mechanical arm. Earlier this year, researchers at the Rehabilitation Institute of Chicago announced that they were testing a

thought-controlled leg. This device works by implanting a metal electrode onto the remaining bone near the amputation site. Test subject Zac Vawter says that the robotic prosthetic allows him to walk as he would have before his leg amputation; he does not need to think about the process and did not have to train his body to use the device.

Thought-controlled prosthetic limbs are especially exciting because the device has the potential to function in a two-way system. Not only does the brain send information to the prosthetic limb but potentially — once a line of communication is established between the brain and the device — the device can send sensory information back to the brain. In other words, these prosthetics will likely have the ability to allow the wearer to experience the sense of touch. In the case of the leg prosthesis currently in testing, this is a critical feature that allows the wearer to walk normally without needing to guess where the foot has landed with each step.

For hand prosthetics, implementing a sense of touch will certainly also improve functionality but contains an emotional component, as well. Imagine how satisfying it will be for a soldier injured in battle to reach out and pat his son on the shoulder for the first time and be able to actually "feel" the experience rather than just see it.

Robotics is also impacting the recovery of patients who need physical therapy due to strokes, accidents, and surgeries. The ZeroG is essentially a harness to assist patients with difficulty walking. The experience is similar to walking on the moon; patients can use the device to safely practice sitting, standing, walking, and climbing stairs. The ZeroG also safeguards the therapist against injury that might be caused by attempting to support the weight of a patient with limited mobility.

The ZeroG allows patients to improve balance and gait while taking comfort in the fact that the harness will catch them if they begin to fall. The HEXORR is an exoskeleton designed for use in therapeutic settings. It is mainly designed for individuals who experience weakness on one side of the body due to a stroke. The robotic device can improve range of motion and grip strength in patients.

Dr. Allen Hoffman, professor at Worcester Polytechnic University, developed a brace to allow individuals with muscular dystrophy to develop better movement in their hands. The device is known as an arm orthosis

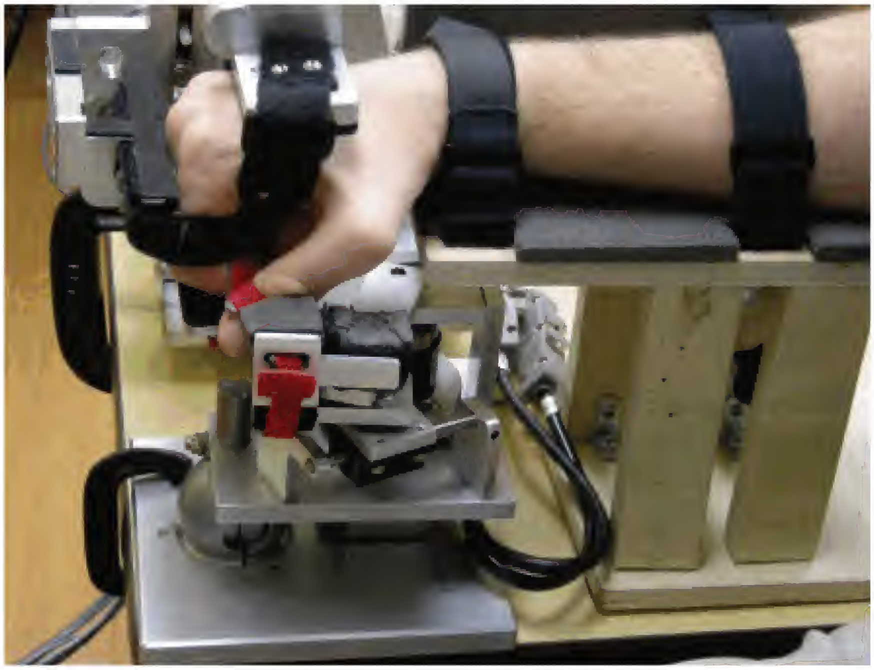
Zac Vawter is able to walk with ease with his thought-controlled robotic leg prosthesis. In this photo, Vawter has just finished climbing 103 stories to the top of the Willis Tower in Chicago, making him the first to do so wearing a robotic prosthetic limb.

Photo courtesy of ABC News.

The i-Limb is one example of the astoundingly sensitive prosthetic hands that are now available for private use.

Photo courtesy of Touch Bionics.

The ZeroG is a robotic harness that allows patients to safely practice walking and climbing. *Photo courtesy of The Catholic University of America.*



The HEXORR exoskeleton is designed to improve strength and range of motion in rehabilitation patients. *Photo courtesy of The Catholic University of America.*

which originates from a Latin word meaning "to straighten." People with muscular dystrophy retain dexterity in their hands, but because of a lack of mobility in the arms and shoulders have trouble using this functionality. The orthosis fits over the arm and is controlled by a joystick held in the free hand. The device allows the wearer to perform simple tasks, such as brushing teeth or eating with a fork. Dr. Hoffman has now patented the orthosis.

I asked Dr. Hoffman how he thought rehabilitation robotics would affect the way patients receive treatment in physical therapy centers. In 5-10 years, he says, physical therapy will be a dramatically different process. "Traditionally physical therapy has involved one or more physical therapists working with a single patient. A considerable portion of the therapy involves the therapist manipulating joints and limbs and using their experience to gather semi-quantitative data to evaluate patient progress over time.

Repetitive motion therapy is used extensively, particularly for persons with stroke," Dr. Hoffman explained. Soon, he says, robotic devices will be found in physical therapy centers across the country. "It will be common for individual physical therapists to control robotic devices which will then provide the motion/resistance to the patient's joints and limbs." The robotic devices will also gather high quality data about the patient's progress during therapy.

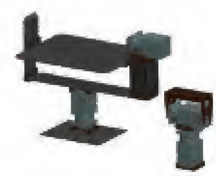
Inside treatment facilities and in the real world, rehabilitation robotics will soon have a major impact on the lives of the disabled. Perhaps before too long, wheelchairs will have been completely replaced by exoskeletons and prosthetic limbs will blend seamlessly into the user's body.

The whole concept of disability will become radically different as affected individuals are able to navigate the world easier than they have ever been able to do. **SV**

The wearable arm orthosis device developed by Dr. Allen Hoffman provides elbow flexion and humeral rotation. It is driven through a wireless connection to a joystick (not shown) operated by the functional hand.

The Actuator Solution for Full Scaled Robots

DYNAMIXEL PRO



Data		Model	H42-20-S300-R	H54-100-S500-R	H54-200-S500-R
		Unit	Data	Data	Data
Rated voltage		V	24	24	24
No load speed		RPM	28.3	35.2	35
No load current		A	0.61	1.06	1.18
Continuous operation	Speed	RPM	15.59	32.7	32.1
	Torque	Nm	5.596	21.142	39.131
	Current	A	1.989	5.930	9.505
Resolution		Step/turn	304,000	502,000	502,000
Gear ratio		-	304	502	502
Backlash		arcmin	3.5	3.5	3.8
Interface		-	RS-485 / CAN	RS-485 / CAN	RS-485 / CAN
Operating temperature		˚C	5~55	5~55	5~55

All in One-Actuator

High Power, High Precision

Up to 200W BLDC motor and up to 500,000 pulse per turn (4096 absolute resolution)

Full Modular Solution

Ease of implementation (no additional design for parts needed)

Ease of maintenance (simple part swapping)

Sophisticated Control Algorithms

Position and speed input commands with dual-loop current control

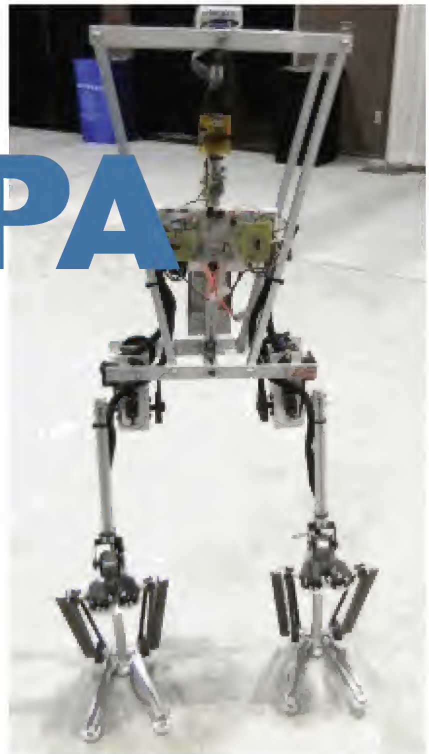
Novel Gear Reduction System

High torque output, light weight, with high impact resistance

USA E-mail : america@robotis.com / Tel : +1-949-333-3635
JAPAN E-mail : japan@robotis.com / Tel : +81-3-4330-3660
Etc. E-mail : contactus2@robotis.com / Tel : +82-70-8671-2609

www.robotis.com

The Road to the DARPA Robotics Challenge



by Daniel Albert

Go to **www.servomagazine.com/index.php/magazine/article/december2013_Albert** to comment on this article.

Part 4: DRC Qualification

We received an email on the October 15th that the qualification requirements to attend this year's DARPA Robotics Challenge had been posted on their website. The new deadline to be invited to run your robot on the test course was at 5 PM EST on October 31st.

It's a good thing we monitored the website and found out about it during the first week in October.

This qualification was in the form of about nine videos of the robot performing all types of tasks. This seems like a reasonable request given that they have spent a great deal of money building robots and funding the teams in Tracks A–C.

We were a bit stressed since (as a Track D entrant) we received no DARPA funds, and additionally we have no corporate sponsorship. We didn't even have our biped walking. Biped designs were not required for these tests that we could easily pass with a rolling robot and one arm.

1. Translate 12 feet.
2. Go over a six inch high concrete block.
3. Pass through a 32 inch doorway
4. Turn a valve on a wall
5. Demonstrate an E-stop switch.

This was a great deal to accomplish in under four weeks!

Deadlines

Well, there is nothing like a deadline to make you productive. We had to work fast. The new servos we ordered from China were delayed in manufacturing. It was almost a certainty they would not arrive in time. Hitec had been advertising some new powered servos and even sent us a sample, but they were not available in quantity either. Robotis also sent us one sample of their new Pro series. Lead time for these was around eight weeks. We needed a minimum of six servos and could not afford to buy five more at the price of more than $2,500 each. What to do?

Yet Another Complete Rebuild

With time running out, we chose the only option available. We had some very strong and very affordable servos from Invenscience. Their Torxis series were priced under $250 each and boasted a whopping 3,200 oz/in each. We had four of these and our initial tests proved that they are monsters. However, they weigh about 1 kilo each (2.2 lbs) and that would create structural problems. A complete rebuild was needed to support the weight

Figure 2.

and torques. This would take most of my waking hours through the month of October.

We already had four of the Torxis. Two were the 3,200 oz/in (model i0600) and two were the 1,600 oz/in. (model i0800). After speaking with Collin at Invensicience, I found out that they also had heavy duty servo brackets for this style servo. I placed a rush order for four more servos and enough brackets to rebuild the complete lower section of the biped.

The New Design

These new heavy servos would not work with any of the previous designs that called for lightweight legs and high powered servos. We quickly came up with a design that Girts could test out in the simulator. We wanted to be able to step over the concrete block, pass through the doorway, and turn the valve. To fully actuate those types of movements usually requires five servos on each leg. We wanted to do it with four. So, we borrowed a trick from the shuffle style bipeds and made really long toes and positioned them sideways. In this manner, we could lift one leg up and still have the center of mass under the foot. (See **Figure 1**.) No side to side tilting is needed.

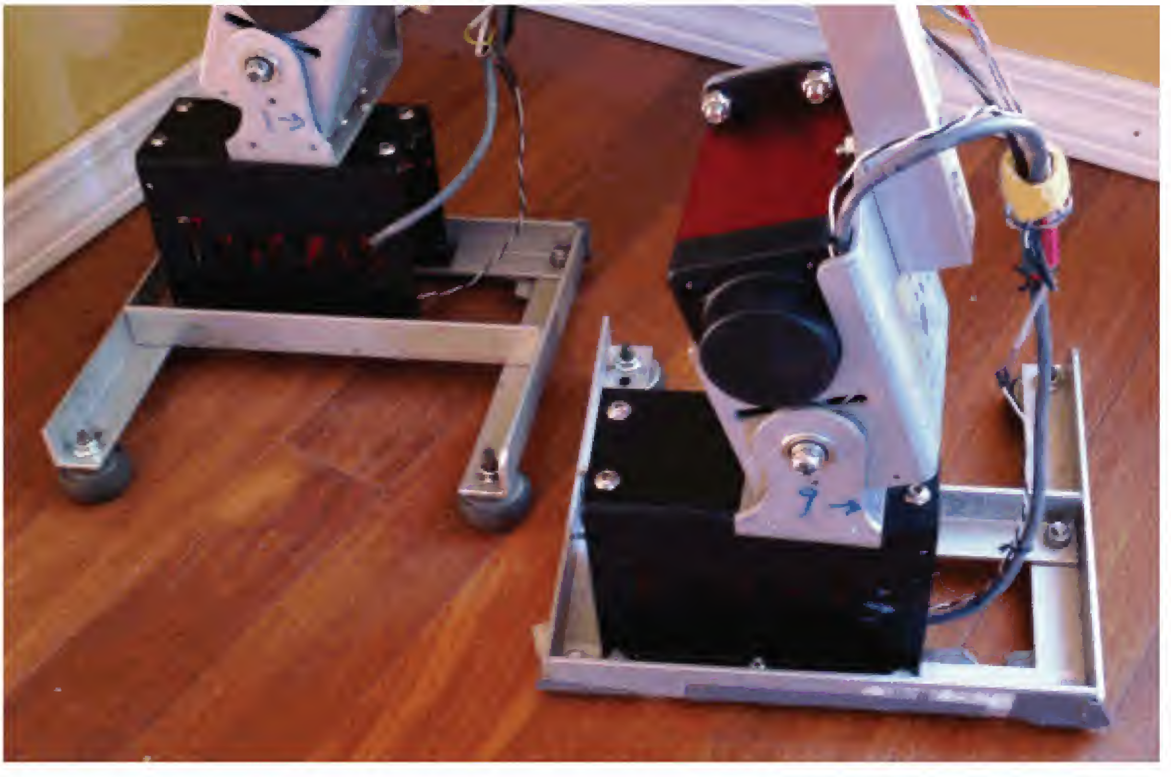


Figure 1.

We took the i0800 servos that didn't have the high torque and made feet that can turn. This enables us to turn the whole body as we walk. We can also bring the foot completely forward from behind without the toes clipping on the opposite leg. The frame sequencer that drives the biped simple turns the long toes backwards during that move so the foot is narrower during that move, and spins them sideways again before the foot lands.

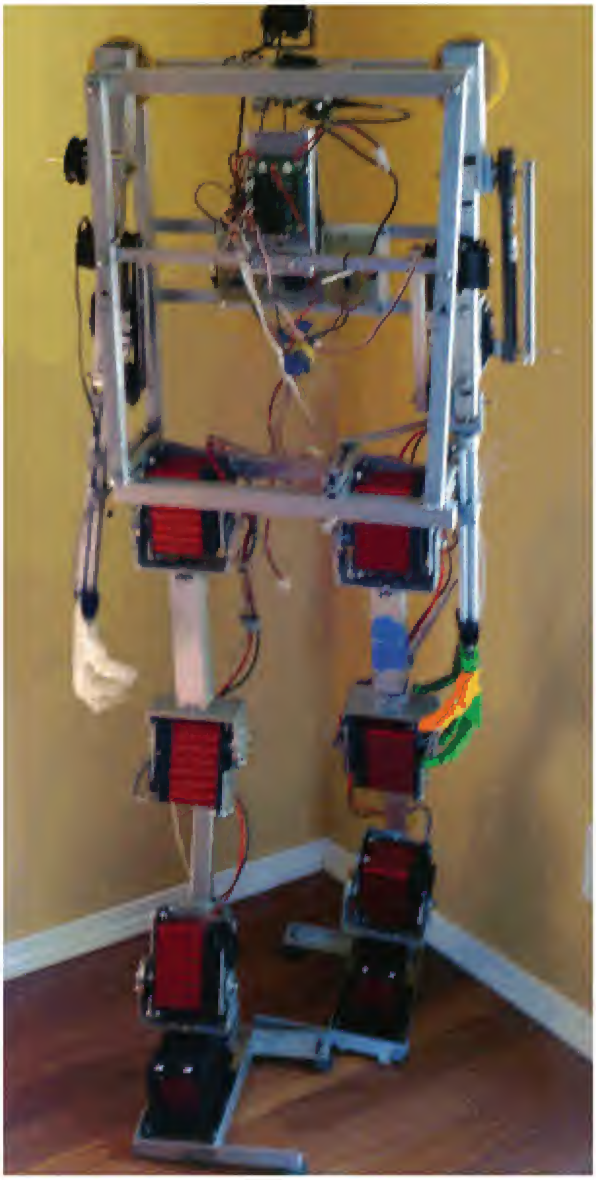
The leg movement is driven by three servos that simulate and ankle, a knee, and a hip that can only move forward and backwards. Unfortunately, after the brackets arrived, I noticed that they were only designed for 90 degree movement and we needed 180 degrees. In addition, since they were not designed for this application there didn't seem to be any way to stack-mount them end-to-end to build a human style leg.

Removing a small piece from the main base of the bracket solved problem one and we found we could get the 180 degrees of movement. Drilling new holes on the back plate and mounting a one inch square aluminum tube to hold two back plates head to toe solved the second problem. (See **Figure 2**.)

We now had a workable platform. The upper torso was not able to support these new servos or the newly designed arms that now had hands built by Glenn and Ansis. So we rebuilt that too. (See **Figure 3**.) Whew!

It was also nice to add a camera with its two-axis pan and tilt to start testing our object-tracking software. Chris is happy to finally have a working test rig for that.

Figure 3.



The Crash

We were all very excited — too excited. We were in a hurry and we rushed the testing. Everything was going so well that we started taking shortcuts. Even though we had WATSON tethered and floating in the air, we made a crucial error and ran the wrong untested sequence. We normally only ran with three people. One person would run the control program; one person would catch the biped if it fell; and one person would be on the E-stop switch. This time, we only had two people.

When we ran the wrong sequence, the servos lurched, the left toe caught the right leg, and the servos just wouldn't quit. We were not fast enough to hit the E-stop and a horrible grinding sound came from one of the gearboxes. Of course, it was the Friday evening before the qualifying videos were due on the following Thursday. With less than a week and without a spare, we were dead in the water for at least three days.

Monday came and I had new parts shipped overnight. Tuesday was mostly spent on rebuilding, and we starting testing the walk that night. There was still two days left.

The First Step

I have been working on this project for seven years, and have gone through at least six full rebuilds. Even though we knew deep down inside we would not meet the deadline, we did not quit. We worked on a static model walk and by Thursday afternoon — about an hour before the deadline — WATSON took its first step. This is the first platform out of six that could successfully repeat a step without falling over.

Two Steppin

My work with smaller bipeds has been helpful, but in many respects things are completely different with larger models. Some things do not scale well. For example, ants. If they were our size they could lift an elephant, but they don't scale well. (Except in cheesy movies). Biped robots are the same.

With the small bipeds, I found it very easy to copy the sequence of the left leg step and reverse it for the right leg to complete a walk cycle. Small differences in the mechanical structure didn't factor that much in the movement. Yet, that doesn't work with larger structures.

We reversed the one-step sequence and found we had to tweak almost every frame. In one part of the step, we needed to add an intermediate frame to retain balance.

It is the Sunday after the qualifications were due, but we didn't give up. WATSON took two full steps and returned to the frame from which we started.

The DARPA Robotics Challenge

So, what of the challenge? Are we out of the competition?

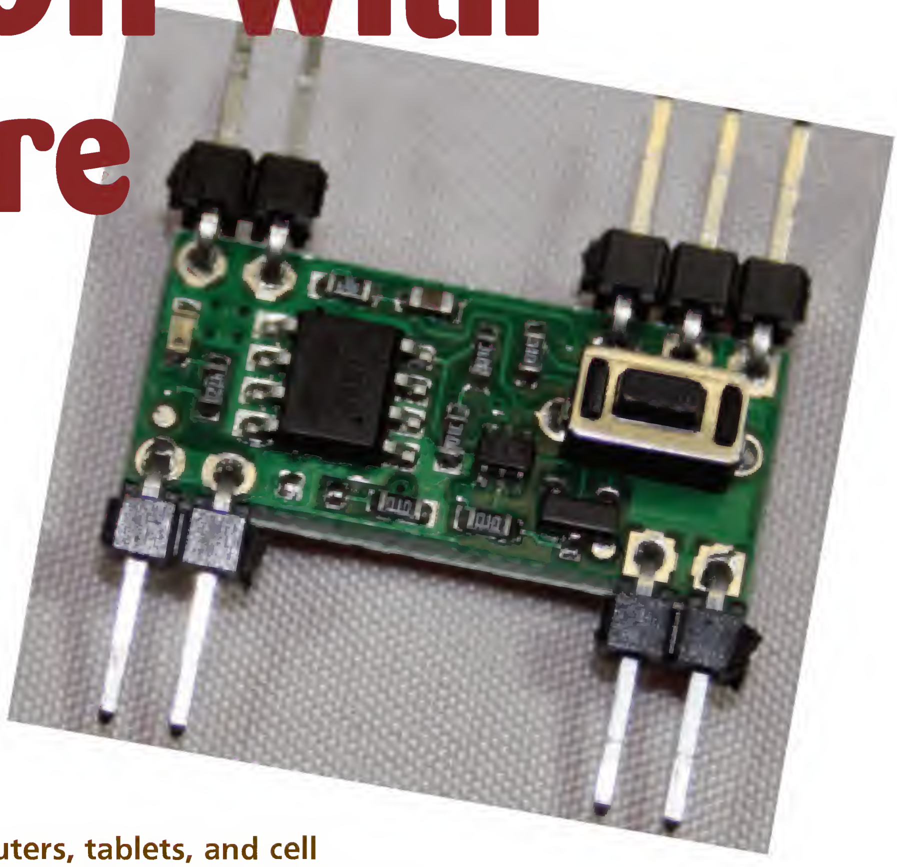
As it turns out, the short answer is ... yes and no.

This year's DRC in December is mostly for Tracks A, B, and C. Since DARPA is footing their bill, the folks running the challenge mostly want to find out how well some of their sponsored teams are performing. So, if we were far enough along, we would have been invited to test WATSON along with Atlas and some other terrific works of engineering. However, Team Walk Like A Man is still entered in the challenge that takes place in December 2014.

Now that WATSON can walk, things will progress even faster. We are still in the running for some of that $2,000,000.00 prize money. SV

Be sure and view the video of WATSON walking (available at the link article).

Turning Your Robot Off With Software



By Roger Tang and Dick Swan

Post comments on this article and find any associated files and/or downloads at **www.servo magazine.com/index.php/magazine /article/december2013_Swan.**

Many electronic items — like computers, tablets, and cell phones — have a power-saving feature that puts the device into a low power state. I've frequently thought that this would be a cool feature to have in my electronics and robot designs; it can be really frustrating when I forget to turn the power off and the battery dies. Fortunately, Pololu makes a nifty little solid-state switch that exactly fits this requirement.

Pololu's device is a solid-state power switch controlled by a momentary pushbutton switch. Tap the button and the switch turns on. Tap it again and the switch turns off. There's nothing unusual about that. Where it gets interesting is that it also has an input pin that can be controlled from your robot's CPU. When a high signal is applied to this pin (labeled "OFF"), the switch turns off. Wiring is extremely simple. Just connect the OFF input to an output pin on your robot's CPU.

Typical Application

There are lots of applications where you might want to automatically power-off your electronics. For this article, we added auto power-off to an existing Arduino-based wireless remote control. The application keeps track of changes to the buttons and joysticks on the remote control. If there's been no changes after a few minutes, then the application will power-off the remote.

There are two versions of the power switch. One is for input voltages in the range 2.5-7 volts and the other is for 4.5-20 volts. Both support a maximum current of 10 amps. The 4.5-20 volt version should be used with an Arduino.

The switch only works for DC since the circuit uses a unidirectional MOSFET to control the current. You must make sure to properly connect the input source polarity.

The software is simple and shown in **Listing 1**. The logic is simple:

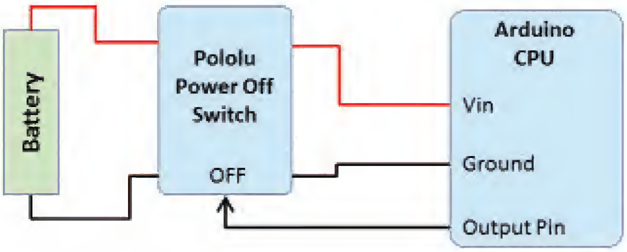
• Every time there's a change to a button or joystick value on the remote, reset a two minute timer.

• If the timer ever expires, then the remote control is inactive and it's time to power-off the device.

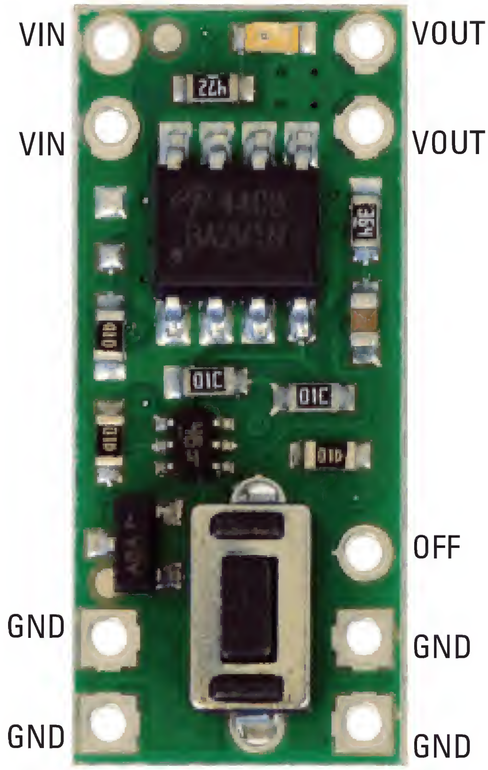
• One minor issue resolved during testing was that the timer was initially stored in a *int* variable which on the Arduino is a 16-bit variable. The circuit kept turning off just after a minute elapsed; likely because 16-bit variables have a maximum value of 65,767 — or 65 seconds with a millisecond timer — before they overflow and wrap around to zero. Changing to a *long* variable resolved this.

Conclusion

The Pololu power switch is inexpensive (under $5) and easy to add to any Arduino robot. It's a good solution whenever you want to be able to provide a software controlled power-off for your robot. **SV**



Wiring/connection diagram.



Pololu pushbutton power switch.

Listing 1.

```
const int offPin = 7;
  // The pin connected to the power OFF function

unsigned long nPowerOffTime;

void resetPowerOffTimer()
{
  // This function resets the time when CPU
  // should be powered off
  // If there is activity on the robot, then
  // call this function to reset the time to
  // power off
  const int kInactivityPeriodForPowerOffSeconds
  = 120;
  nPowerOffTime = millis() + (kInactivityPeriod
  ForPowerOffSeconds * 1000);
  return;
}

void setup()
{
  // In Arduino, the "setup" function is called
  // once when user program starts

  // Setup the "offPin" to be an output. Initial
  // value is low.
  pinMode(offPin, OUTPUT);
  digitalWrite(offPin, LOW);

  resetPowerOffTimer();
  return;
}

void sendRemoteControlMsgToRobot()
{
  ...Send Wireless message to remote
  containing joystick and button values...
  if (...Joysticks or buttons have changed
  value...)
    resetPowerOffTimer();
}

void loop()
{
  // In an Arduino application, the loop
  // function is called continuously.
  // The user's application should do its work
  // and return. Between calls to to the "loop"
  // function other housekeeping functions are
  // performed.

  int nCurrTime = millis();

  {
    // Every 50 msec send a wireless message to
    // robot with joystick values

    const int kTimeBetweenRCMessages = 50;
    static int nTimeForNextRemoteControlMessage
    = 0;
    if (nCurrTime >= nTimeForNextRemote
    ControlMessage)
    {
      nTimeForNextRemoteControlMessage =
      nCurrTime + kTimeBetweenRCMessages;
      sendRemoteControlMsgToRobot();
    }
  }

  // Check to see if CPU should be automatically
  // powered off
  {
    if (nCurrTime >= nPowerOffTime)
      digitalWrite(offPin, HIGH);
  }
}
```

Build the Plastic Bot of Destruction

Part 2

By Michael Simpson

Post comments on this article and find any associated files and/or downloads at **www.servo magazine.com/index.php/ magazine/article/december 2013_Simpson**.

This time, I'll go over the wheels, gears, and chassis parts for our Plastic Bot of Descruction (PBOD). The STL drawing files for the printed parts are available on the Kronos Robotics website. I will list links later in the article.

Figure 1.

Wheels

The six wheels shown in **Figure 1** consist of three parts each:

- Wheel Hub Cap
- Wheel Geared Hub
- Wheel Tire

Wheel Hub Cap

The hub cap shown in **Figure 2** is a 3D printed part. It consists of three posts that are meant to insert into mounting holes in the wheel tire. The holes in the posts are .089" in diameter and should be tapped with a #4-40 tap. The overall diameter is 1.7", so you should be able to print four or more on most 3D printers. If you are printing in PLA, it is highly recommended that you print at least three wheels to keep the posts from overheating and melting.

Wheel Geared Hub

The geared hub shown in **Figure 3** is driven by either the motor or idler gears. It is attached to the hub cap and tire assembly via four #4-40 x 3/8" machine screws. While you can print more than one of these at a time, it is recommended that you print only one at a time to keep the gear as accurate and clean as possible.

Figure 2.

Figure 3.

Figure 4.

Figure 5.

Figure 6.

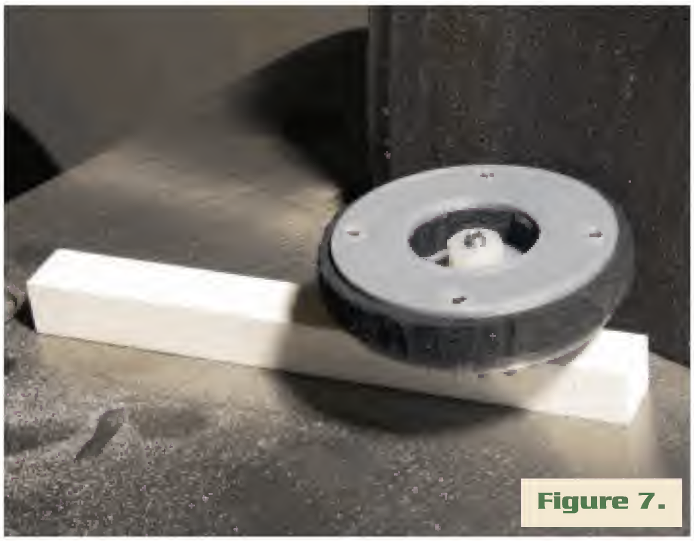


Figure 7.

Wheel Tire

The tire shown in **Figure 4** is made from 1/4" craft foam. Originally, I was going to provide a set of laser-cut tires. Unfortunately, my laser is out of commission for the time being, so I used my new KRmc01 CNC mini mill to make them. For now, you will have to make your tires manually by cutting a 1.85" circle out of 1/4" craft foam. Use the wheel gear hub centered on the tire and either mark or drill the four holes. Enlarge the holes to roughly 3/16". The center cutout is .8" in diameter, but its size is not critical.

Figure 8.

Figure 9.

Figure 10.

Figure 11.

Figure 12.

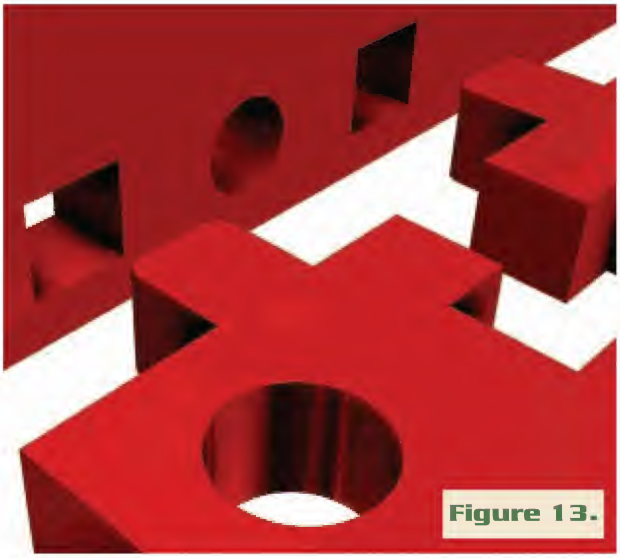


Figure 13.

Figure 14.

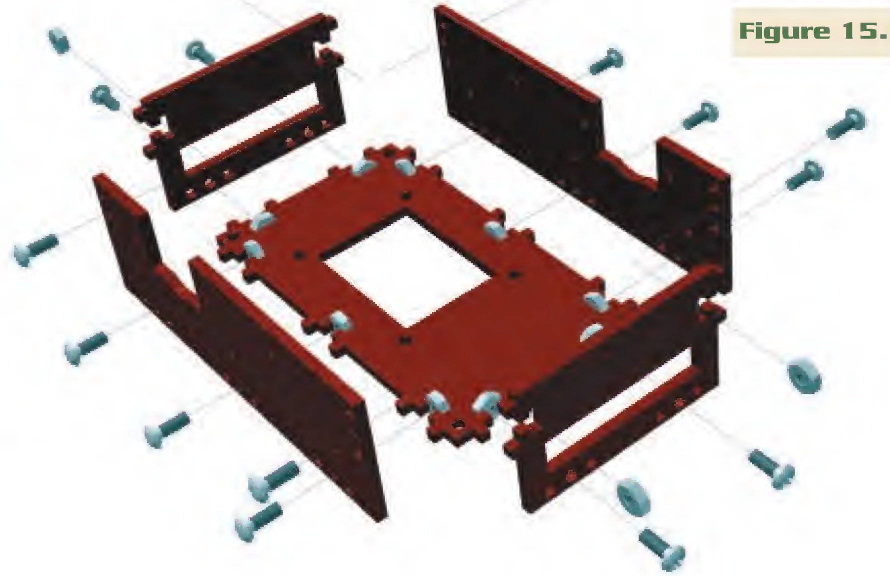


Figure 15.

Wheel Assembly

Slip the tire over the hub cap as shown in **Figure 5**. Place the geared hub on top of the assembly and secure it with four #4-40 x 3/8" machine screws as shown in **Figure 6**. The easiest way to clean up the tire and make it run true is to mount the tire to a piece of scrap and slowly rotate it against a sander like the one shown in **Figure 7**. You will need six wheels for the PBOD. Once complete, set them aside.

Gears

There are four gears used to put the six wheels in motion: two idler gears and two drive gears. One each is used on each side of the PBOD.

Idler Gears

The idler gear shown in **Figure 8** is positioned between two wheels on each side of the PBOD as in **Figure 9**. You will need to print two idler gears.

Drive Gears

The drive gear (**Figure 10**) is attached to the motor and is used to drive two wheels. To attach the gear to the motor shaft, you will need two #2-56 hex nuts and two #2-56 x 1/4" machine screws. The hex nuts are inserted into the two slots in the drive gear and secured with the two screws as shown in **Figure 11**.

You will need two drive gears — one for each motor.

Chassis

The PBOD chassis is made up of the six components shown in **Figure 12**. These components are the sides, bottom, and ends. In addition, there are two motor mounts. Printing the parts can be problematic when printed in ABS as they may want to warp. They can also be difficult to remove from the bed. If you have problems printing them in ABS, try switching to PLA on heated glass.

Basic Chassis Assembly

Construction of the assembly is relatively simple. One part is slipped over the small cogs on another part as shown in **Figure 13**. If the fit is too tight, you can use a razor knife or file to reduce the size of the cogs.

Once attached, a #6 hex nut is inserted into the slot as in **Figure 14**. A 6-32 x 3/8" machine screw is used to secure the nut and hold the parts in place. **Figure 15** shows a complete exploded view of the chassis. While I recommend nylon hardware for this portion of the assembly, you can use steel or aluminum hardware. You can also add a small #6 lock washer to the machine screw to keep them from working loose. This works for both metal and nylon hardware.

Conclusion

You can find the STL files for the various parts on the PBOD web page at **www.kronosrobotics.com/pbod**.

One thing you need to know, however, is I had placed four small holes in the corner of the chassis bottom so I could install a top to the chassis that would hold the weapon. The problem is the hardware for the wheels would not allow me to place either a standoff or spacer in the corners. For this reason, I decided to add some small tabs at the top of the sides and end parts to secure the chassis top.

Next month, I will complete the chassis assembly and add the wheels, gears, motors, controller, and radio. Once complete, you will be able to take the PBOD for a spin, literally. Please be sure to post questions and suggestions in the *SERVO Magazine* forums at **http://forum.servomagazine.com/viewtopic.php?f=49&t=17029.**

I am still looking at machining the parts on a CNC mill for you folks that don't have 3D printers. The chassis shown at the beginning of the article was actually machined with my CNC mini mill. **SV**

Windows 8 Tablets

By John Blankenship and Samuel Mishal

Post comments on this article and find any associated files and/or downloads at **www.servomagazine.com/index.php/magazine/article/december2013_Blankenship**.

Most hobby robots are powered by a small microcontroller with limited capabilities. The new Windows 8 tablets provide far more power with an impressive array of internal sensors, text-to-speech, and voice recognition — all contained in a relatively small lightweight form that makes them a real contender for your next robotic project. Are these tablets the ultimate robot controller? This series of articles hopes to convince you they are.


Figure 1.

Have you ever wished you could control your robot with a full sized high speed computer instead of a small microcontroller? Imagine having the memory and power to implement voice recognition and computer vision. Consider the computational advantages of floating-point math and multidimensional arrays. Imagine how human interaction might be improved if your robot had all the capabilities of a PC.

Most hobby robots are not large enough to house a desktop machine or even a powerful laptop, but the size and weight of the machine itself is not the only consideration. A large computer will require a larger battery, and that extra weight means larger motors. That, in turn, requires an even larger battery. Such considerations have limited the computing power available for small robots, but the new Windows 8 tablets now offer a potential solution.

Many computer manufacturers offer these tablets with the power of a desktop in a form only slightly larger than some Android tablets. **Figure 1** shows a Samsung Ativ 700t tablet mounted on an RB-9 robot (available from

The Ultimate Robot Controller Part 1

www.RobotBasic.org). As you can see, just displaying a face on the tablet can give your robot a little personality. Future articles in this series will discuss the mounting of the tablet, as well as how to utilize the text-to-speech capabilities of Windows 8 to give the robot a voice that is synchronized with the face's mouth movements.

If you are thinking of purchasing a tablet to control your robot, the specifications can vary greatly, so it is important to consider your particular needs. For example, some models sacrifice speed in exchange for lighter weight or longer battery life, but remember, even the *slowest* models are far more capable than the microcontrollers typically used in most hobby robots.

Since most Windows tablets support a standard keyboard docking option, they can also be used as a development station. This eliminates the need to write code on one machine and download it to another for testing — a feature that can reduce development time considerably. For many people, this feature alone would be worth using a tablet as a robot controller.

While size, weight, and speed are important considerations, Windows 8 tablets offer so much more. All models come with three-axis gyroscopes, accelerometers, and magnetometers. If you have worked with such sensors, you know it can take some serious mathematics to blend their outputs into usable data. Microsoft's Fusion technology handles all that for you, creating both a tilt-compensated compass and a virtual inclinometer with amazing stability and accuracy.

Additionally, all Windows 8 tablets have one (and sometimes two) camera, an ambient light sensor, and standardized internal firmware for interfacing with compatible USB GPS units. Adding all these tablet features to the standard sensors often found on hobby robots can produce a robot with amazing capabilities. Accessing the internal sensors, though, generally requires using Window's 8 programming tools for Metro-Style-Apps — tools that might not be your first choice for programming a robot.

In order to make the tablet's sensors available to a wide variety of desktop-style programming languages, we designed a Sensors Interface Utility program and contracted Windows' developer Shobhan Taparia to create an initial shell program for us. This gave us an immediate way to begin experimenting, and provided us a basic framework that we could modify and improve upon.

During the design phase, we had to decide how our utility program would communicate with the desktop application. We needed an approach that was relatively fast and reliable, yet one that could be used with a wide variety of languages, including our preferred language — RobotBASIC (a free language that can be downloaded from **www.RobotBASIC.org**).

Since most modern languages have functions for dealing with the clipboard, we decided that the Sensors Interface Utility would communicate with primary control programs using the clipboard. For example, if a program wants to read the sensors, it would simply put the command word *sensors* on the clipboard and the utility program should respond by filling the clipboard with the sensor data in a specific format.

The only problem this scheme has is that both the utility and the application are constantly monitoring the clipboard. Since the clipboard is generally accessed manually for copy-and-paste operations, it appears that Microsoft never worried about two programs accessing it simultaneously, because our tests showed that such an action can cause errors in either program.

We solved this problem by developing a hand-shaking arrangement using a simple text file. When the file's name is *SensorsTurn.txt*, the utility program knows it should look at the clipboard for a command. After placing the requested data on the clipboard, the utility renames the file to *RBsTurn.txt*, telling RobotBASIC (or another programming language of your choosing) that the data is ready to be accessed.

The above process might sound complicated, but it is actually very easy to implement as shown in the RobotBASIC program in **Figure 2** which obtains and displays the tilt and compass data using the utility program.

The program is short because RobotBASIC has many high level commands to perform actions such as renaming files and parsing strings. If you wish to use a different language, use the comments in the program to understand what is happening and duplicate the actions with the commands available in your language.

The commented program should be easy to follow, but we will still need a short discussion of how the clipboard data is formatted. The data placed on the clipboard by the

```
Main:
   call InitSensors(0,0,0,90) // set offsets
   call Orientation(Forw,Side,Rot,Heading)
   // display the entire array
   for i=0 to 6
      xyString 10,10+20*i,i; Sensors[i]
   next
   // display the processed readings
   xyString 10,160,"Forward Tilt     = ",Forw
   xyString 10,180,"Side Tilt        = ",Side
   xyString 10,200,"Rotational Tilt = ",Rot
    xyString 10,220,"Compass Heading = ",Head-
ing
end

sub InitSensors(a,b,c,d)
  dim Sensors[28] //max available from utility
  for i=0 to 27
    Sensors[i]=""
  next
  // allow setting certain angles to zero
  dim Offsets[4]
  Offsets[0]=a
  Offsets[1]=b
  Offsets[2]=c
  Offsets[3]=d
return

sub Orientation(&t1,&t2,&t3,&h)
   // Request readings
   SetCBtext("tilt")
   filerename ("RBsTurn.txt","SensorsTurn.txt")
   // and wait for it to respond
   while !fileexists("RBsTurn.txt")
   wend
   // delay 10
   // get the data
   s=getcbtext()
   // and parse it
   mfromstring Sensors,s,","
   t1 = round(Sensors[1])+Offsets[0]
   t2 = round(Sensors[2])+Offsets[1]
   t3 = round(Sensors[3])+Offsets[2]
   h  = round(Sensors[5])+Offsets[3]
   // ensure heading within normal range
   if h>359 then h-=360
   if h<0 then h+=360
return
```

Figure 2.

utility program is a single string with each item in the string separated by a comma. To make it easier to deal with the data, headings are placed in the string that describe the data within the string itself. Let's look at an example.

When the tablet's orientation data is requested (using the command *tilt*), the first item in the string is the heading *Inclinometer* with the next three items being the data indicating the current tilt angles. Following the angle data is the heading *Compass*, which is followed by the two data parameters. The first reading is for magnetic north, followed by the true north reading if it is available. In order to make this data more easily accessible, we created two functions: one called *InitSensors()* and the other called *Orientation()*. The *initialization* function dimensions two arrays, and fills one of them with the four offset values that are passed to it. More on these offsets shortly.

0	Inclinometer
1	90.04
2	-3.34
3	270.34
4	Compass
5	89.6799979954958
6	
Forward Tilt	= 90
Side Tilt	= -3
Rotational Tilt	= 270
Compass Heading	= 180

Figure 3.

The *Orientation* function handles all the communication with the clipboard and parses the data string obtained from the sensor utility, placing the individual data elements into the array *Sensors[]*. The individual elements of this array can be accessed directly to obtain the desired data but we wanted to make using the data even easier. In our example, the tilt readings and compass heading are truncated to integers and offset by the amounts specified during initialization before being placed into the variable names passed to the function.

You will notice a 10 ms delay has been commented out in the function just before the data is retrieved from the clipboard. Windows performs clipboard operations in the background, so it is possible that some systems might need a small delay to ensure the data is ready. If your system encounters an error while accessing the clipboard, add a small delay at this point.

Our function provides integer readings because they are often easier to use, but you can keep the floating-point format if such accuracy is needed for your projects. The offsets for the tilt parameters allow you to create a "normal" orientation for your tablet. For example, the forward tilt reading is normally zero (at least on the tablet we used) if the tablet is lying flat like a piece of paper on a desk. If your tablet is going to be used in a vertical position, then applying an offset of -90 to the forward tilt parameter can make the data more logical because a zero reading would then mean the tablet is in a normal upright position.

Having the ability to add an offset to the compass reading can also be valuable. Our tablet, for example, gives a due north reading when the back of a vertical tablet

faces north which is 180° out of phase with what you would want if the tablet is mounted on a robot as shown in **Figure 1**. As another example, it is not inconceivable that someone might prefer to think of north as a specific wall in the room where your robot operates. Since the compass reading should always have a value between 0 and 359, the offset value is so adjusted, if necessary, before placing it into the designated variable.

Figure 3 shows the data displayed when the program of **Figure 2** is run. The top of the figure shows the parsed data displayed directly from the *Sensors[]* array. The bottom of the figure shows the modified data displayed using the variable names passed to the *Orientation* function.

Before you execute the program shown in **Figure 2**, you must ensure that the Sensors Interface Utility is running. The utility program and its handshaking text file should reside in the same directory as the application program unless you add path names to all the associated commands.

If you are using RobotBASIC, you have the option of setting the default directory either through the FILE menu or within a program using the *DirSet()* function. We assumed that some users might prefer to have their application start the Sensor Utility, so we added support for this option by having it terminate if another copy is already running. You can run the utility from RobotBASIC with the command *Spawn ("RBsensorUtility.exe", P_NOWAIT)*.

Since the Sensor Utility runs entirely in the background, we have it automatically start in the minimized mode. If you click the tray icon though, you will see the splash screen as shown in **Figure 4** which has a button for terminating the program. You can also terminate the program through the command interface by placing the word *Stop* on the clipboard and renaming the text file to tell the utility to process the current command.

The Sensor Utility has numerous commands for obtaining the desired sensor data. Placing the command word *Raw* on the clipboard, for example, will request the raw data from the gyroscope and the accelerometer (eight total items including the two headings). The command *BasicSensors* will return all of the readings shown in **Figure 5**, and the command *Sensors* returns that data plus eight more readings associated with the full orientation sensor. There are other commands for obtaining images from cameras and GPS data, but they will be discussed in future articles.

We had problems with the orientation sensors that are worth discussion. On two occasions, the compass heading and the tilt values on our tablet become erratic or zero. This appeared to happen when the tablet was used near a TV or other appliance with a significant magnetic field, but we cannot confirm that as the cause. An Internet search indicated that other Windows 8 tablet owners also had this problem.

Slowly rotating the tablet (without the keyboard) on all three axes fixed the problem for us, but only if no programs were accessing the sensors (so close your utility before performing this procedure). We assume the rotation causes the tablet's firmware to perform an internal calibration procedure, but no one at Samsung tech support could verify this.

Figure 4.

As nice as it is to have access to the internal tablet sensors, most robotic projects will also require some form of proximity and/or ranging sensors, as well as some way to control the robot's motors. This generally means your tablet needs to have a USB connector on the tablet itself and not just on the keyboard. Most Windows tablets do have a USB connector, but even the ones that don't could use the built-in Bluetooth for I/O operations. Let's examine four basic ways of interfacing a tablet with external sensors and motors:

- **USB I/O:** Adding a USB I/O board is a straightforward way of adding I/O pins to the tablet. We have used the U4xx devices (see the U401 in **Figure 6**)

0	Light Sensor
1	6
2	Accelerometer
3	-0.0780000037048012
4	-1.02600004873239
5	-0.0340000016149133
6	Gyroscope
7	0.56000002659857386
8	0.0700000033248216
9	0.07000000332482166
10	Inclinometer
11	88.36
12	-4.59
13	267.61
14	Compass
15	92.4699979331344
16	
17	Simple Orientation Sensor
18	Not Rotated

Figure 5.

Figure 6.

from **USBmicro.com** because RobotBASIC has commands for accessing them directly. USBmicro supports many languages, so if this option interests you, check out their webpage at **www.USBmicro.com**.

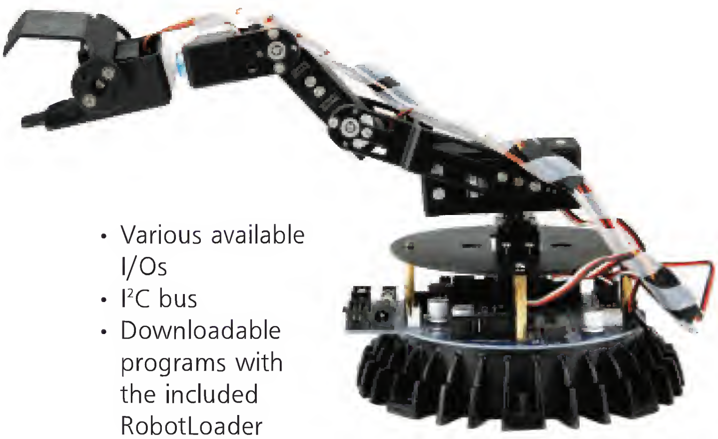
- **Serial I/O:** A USB-to-serial adapter can allow the tablet to communicate with serial I/O boards or even a small microcontroller that you program to serve as your I/O device. We explored this option in the book, *Hardware Interfacing with RobotBASIC*.
- **Bluetooth:** The tablet's built-in Bluetooth can effectively provide a serial link to external devices without requiring a USB port, but you should keep in mind that Bluetooth switching delays between transmit and receive modes can be a bandwidth bottleneck for many situations.
- **RROS Chip:** Our own RROS (RobotBASIC Robot Operating System) offers an ideal solution as discussed in the September 2012 issue of *SERVO*. It provides both the physical interface and the internal firmware necessary to control the motors and read the sensors of a remote robot. It may be connected through a direct USB serial cable or over a Bluetooth link (bandwidth is not an issue in this case because CPU intensive operations — such as motor control — are carried out completely in the background by the RROS chip). Visit **www.RobotBASIC.org** for more information on the RROS chip.

The interfacing method you choose can depend on which tablet you have and the nature of your application. Future articles in this series will provide example programs that utilize the tablet's orientation sensors, as well as its camera and a USB GPS. Seeing examples of how the sensors can be used will help you determine what tablet and interfacing methods might be right for you.

Next time, we will examine two applications to demonstrate how easily a tablet's sensor data can be used to control a robot. If you just can't wait till next month and would like to start experimenting on your own, you can obtain a copy of the Sensor Interface Utility program by requesting it with an email to RobotBASIC@yahoo.com. Please use *Sensor Interface Program* as the subject of the email to speed the response. **SV**

Continued from page 17

R680 Banshi Robotic Arm



Global Specialties introduces their Banshi robotic arm. The R680 is an affordable robot for educators and hobbyists. It is ideally suited to learn the basics of electronics, mechanics, and programming. The Banshi is controlled by a powerful ATmega64 microcontroller that is programmable via open source tools in C.

The robot comes with many example programs already written that can be downloaded to the robot using the supplied USB interface and the RobotLoader software. Or, users can write their own custom programs using the free open source WinAVR software. The robot arm can be controlled using the included keyboard or RACS software. Using the software, movements of the Banshi can be recorded and played back.

The I/Os (inputs and outputs) — together with the flexible I²C bus system — allow the addition of extra modules, thus enabling the robot to react to its environment. The Banshi Robot features:

- Six degrees of freedom
- ATmega64 processor
- Complete 72 page manual
- USB interface
- Various available I/Os
- I²C bus
- Downloadable programs with the included RobotLoader software
- Easy control with keyboard or RACS software
- Plastic arm and metal chassis
- 12V power supply

The Banshi comes unassembled as a kit. Assembly tools are included. The Banshi robotic arm lists at $199.

Global Specialties

Website: **www.globalspecialties.com**

The SERVO Webstore

Attention Subscribers ask about your discount on prices marked with an *

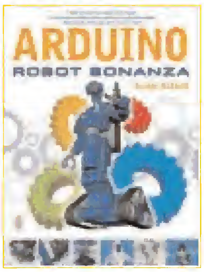
CD-ROM SPECIALS

12.2013 FEATURED

ROBOTICS

Arduino Robot Bonanza

by Gordon McComb



Create high-tech walking, talking, and thinking robots. "McComb hasn't missed a beat. It's an absolute winner!"

Breathe life into the robots of your dreams — without advanced electronics or programming skills. *Arduino Robot Bonanza* shows you how to build autonomous robots using ordinary tools and common parts. Learn how to wire things up, program your robot's brain, and add your own unique flair.

$29.95*

Mechanisms and Mechanical Devices Sourcebook 5th Edition

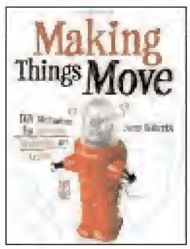
by Neil Sclater

Fully revised throughout, this abundantly illustrated reference describes proven mechanisms and mechanical devices. Each illustration represents a design concept that can easily be recycled for use in new or modified mechanical, electromechanical, or mechatronic products. Tutorials on the basics of mechanisms and motion control systems introduce you to those subjects or act as a refresher.

Reg $89.95

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts



In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

$29.95*

Build Your Own Humanoid Robots

by Karl Williams

GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot.

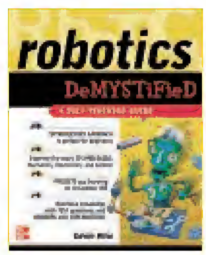
$24.95*

Build Your Own Teams of Robots with LEGO® MINDSTORMS® NXT and Bluetooth®

Plan, design, assemble, and program robot squads that communicate and cooperate with each other to accomplish together what they can't do individually. Inside the authors show you how to construct a team capability matrix (TCM) and use the Bluetooth Robotic-Oriented Network (BRON) so your robot teams can share sensors, actuators, end effectors, motor power, and programs. **$30.00**

Robotics Demystified

by Edwin Wise



YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

Or order online
www.servomagazine.com

SPECIAL OFFERS

Forbidden LEGO

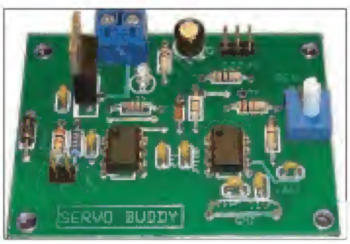
by Ulrik Pilegaard / Mike Dooley

Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg $24.95 Sale Price $19.95

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.

For more information, please check out the May 2008 issue or go to the SERVO webstore.

Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

3D LED Cube Kit

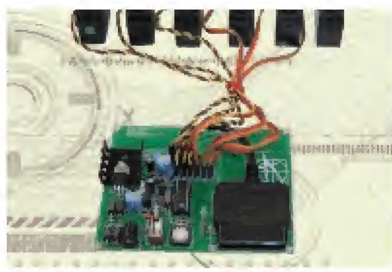
From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of *Nuts & Volts Magazine*.

This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes.
Colors available: Green, Red, Yellow & Blue.
Jig and plastic cases also available.

Subscriber's Price **$57.95**

Non-Subscriber's Price **$59.95**

PS2 Servomotor Controller Kit

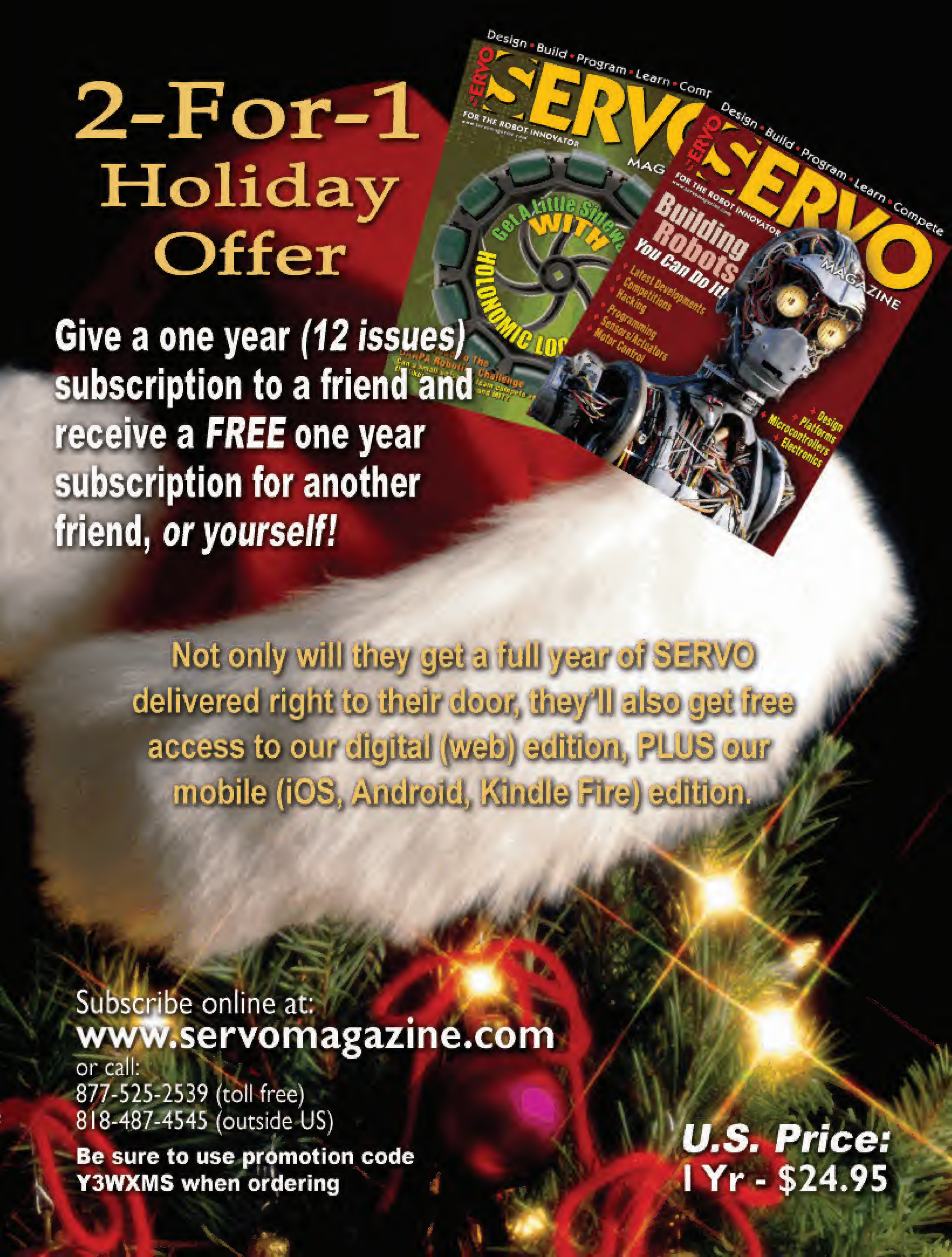


This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.

For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price **$79.95**

Non-Subscriber's Price **$84.95**


2-For-1
Holiday
Offer
Give a one year (12 issues) subscription to a friend and receive a FREE one year subscription for another friend, or yourself!
Design • Build • Program • Learn • Compete
SERVO
FOR THE ROBOT INNOVATOR
MAGAZINE
Building Robots You Can Do It!
Latest Developments
Competitions
Hacking
Programming
Sensors/Actuators
Motor Control
Design
Platforms
Microcontrollers
Electronics
Not only will they get a full year of SERVO delivered right to their door, they'll also get free access to our digital (web) edition, PLUS our mobile (iOS, Android, Kindle Fire) edition.
Subscribe online at:
www.servomagazine.com
or call:
877-525-2539 (toll free)
818-487-4545 (outside US)
Be sure to use promotion code Y3WXMS when ordering
U.S. Price:
1 Yr - $24.95

Twin brothers hack whatever's put in front of them, then tell you about it.

by Bryce Woolley and Evan Woolley

Go to **www.servomagazine.com/index.php/magazine/article/december2013_TwinTweaks** to comment on this article.

To Infinity and Beyond

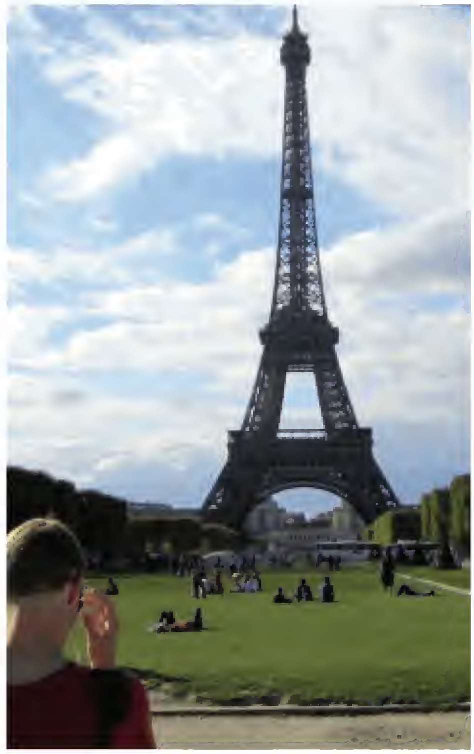
Few things capture the imagination like robotics. The idea of a robot may conjure up images of everything from giant arms making cars to autonomous Iron Man suits. One other endeavor has the same power as robotics to galvanize the imagination from science fiction to science fact — space.

Both present promises for great adventure, and combining the two is perhaps the greatest adventure of all. Just think of those tenacious rovers Spirit, Opportunity, and Curiosity, and how those appropriately named robots are the cutting edge of progress and inspiration.

Robotics and space are the perfect combination because nowadays we let robots blaze the trail before we let a person boldly go where none has gone before.

With that in mind, this month we wanted to explore a challenge that sees a robot as the trailblazer into the wild blue yonder — the space elevator.

THIS IS HOW WE GET TO SPACE ...



LET'S MAKE THAT — JUST 110,000 TIMES TALLER!

Cat's Cradle

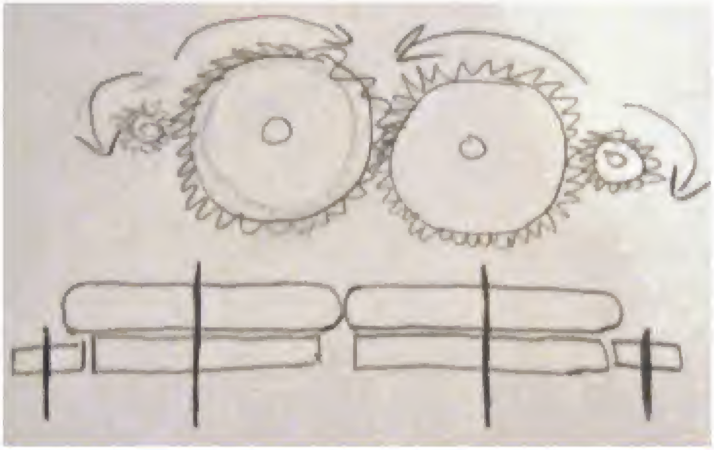
Though many anecdotal accounts of apple-falling, head-bopping scientific inspirations may be apocryphal, we were reminded of the space elevator by an unlikely source. Our cat Lida loves to bat around ribbon, and often seems dead set on the idea of pulling it out of our stronger grasp.

When she's feeling particularly feisty, it's almost as if she wants to hoist herself onto the ribbon if that is what is necessary to claim victory.

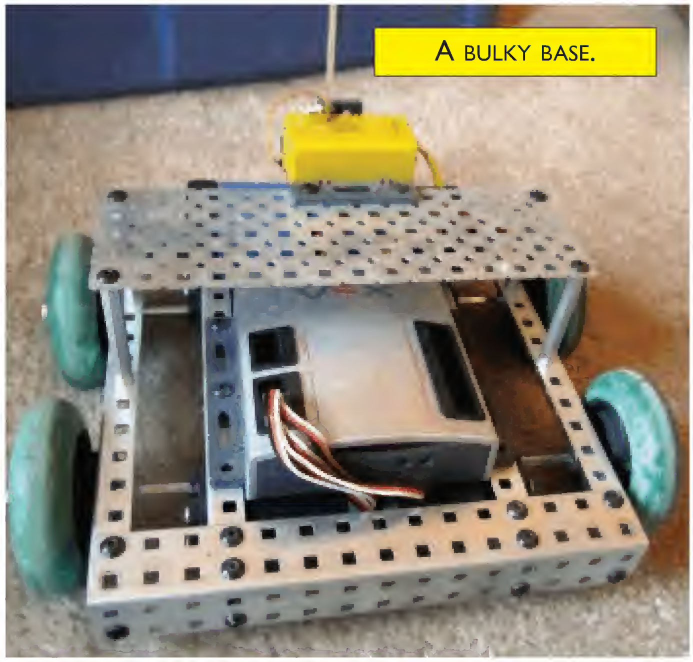
That reminded us of the way that the space elevator problem is often presented at robotics competitions — the challenge to build a ribbon climber.

The space elevator, however, was not actually inspired by a playful feline. The idea for the space elevator was first published by Russian rocket scientist Konstantin Tsiolkovsky way back in 1895. Tsiolkovsky was initially inspired by the literature of Jules Verne, and his specific inspiration for the space elevator appears to have been the Eiffel Tower which was completed in 1889.

As the tallest structure in the world when it was completed and meant as a testament to humankind's innovative ability, the Eiffel Tower certainly electrified many imaginations. Tsiolkovsky imagined a structure spanning from the surface of the Earth to the height of geostationary orbit, about 35,790 km. Tsiolkovsky's inspiration — the Eiffel Tower — reaches a height of only about a third of a kilometer. The problem with Tsiolkovksy's vision is that it was for a tower built kind of like the Eiffel Tower



SKETCHING OUT A DESIGN.



A BULKY BASE.

— a compressive structure supported from the bottom up.

No known material, however, has the compressive strength to support a free-standing tower tall enough to reach geostationary orbit. Other scientists, however, picked up where Tsiolkovsky left off.

The modern proposal for a space elevator is to deploy a cable or tether (or ribbon, if you will), from a satellite down to Earth. A counterweight on the tether in space means that the structure is under tensile stress instead of compressive stress. Even though we're talking about serious tensile stress, the tensile structure deals with forces much more manageable than a compressive structure about as tall as 110,000 Eiffel Towers.

Nowadays, we have a material that makes such a structure feasible: carbon nanotubes. With an achievable structure within reach, the space elevator has been the subject of a renewed research effort, and detailed proposals have emerged that fill in some of the important specifics about the project such as location, space debris avoidance, and other head scratchers.

One proposed space elevator would climb at what sounds like a speedy 200 km/h. That means that the trip to geostationary orbit would only take a mere seven and a half days. That's a long time to listen to *The Girl From Ipanema*.

One way in which the space elevator has been pushed closer from the realm of science fiction into the realm of science fact is through competition. Much like the Ansari X-Prize, several organizations seek to spur innovation with a monetary incentive. One such organization was Elevator:2010, a partnership between the Spaceward Foundation and NASA Centennial Challenges.

Elevator:2010 sponsored competitions for five years beginning back in 2005, and presented two different challenges. One was to build the strongest tether possible. The other was to create a wirelessly powered climber capable of ascending a tether at a certain speed.

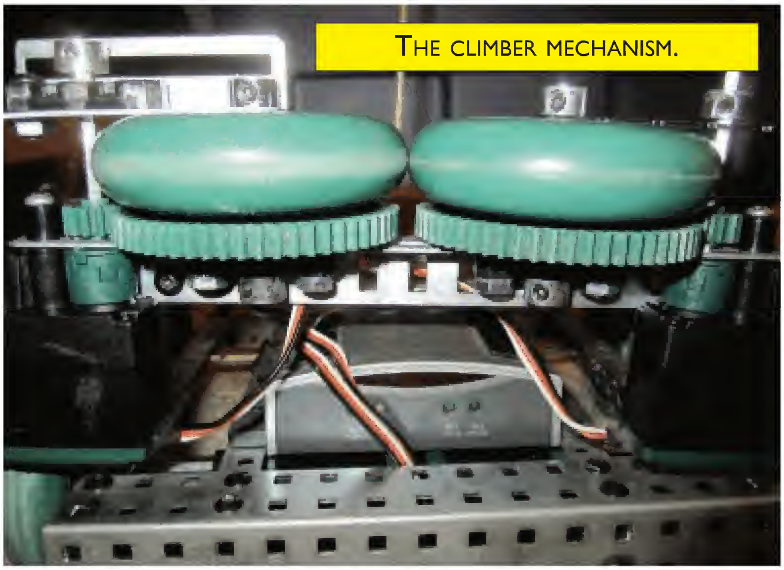
RoboGames also included a ribbon climbing challenge inspired by the space elevator. For their challenge, teams had to build a robot to ascend a tape ribbon up to six meters high, pause in the middle on command, and stop automatically at the top. Bonus points were awarded to teams who beamed power from the ground to their robot.

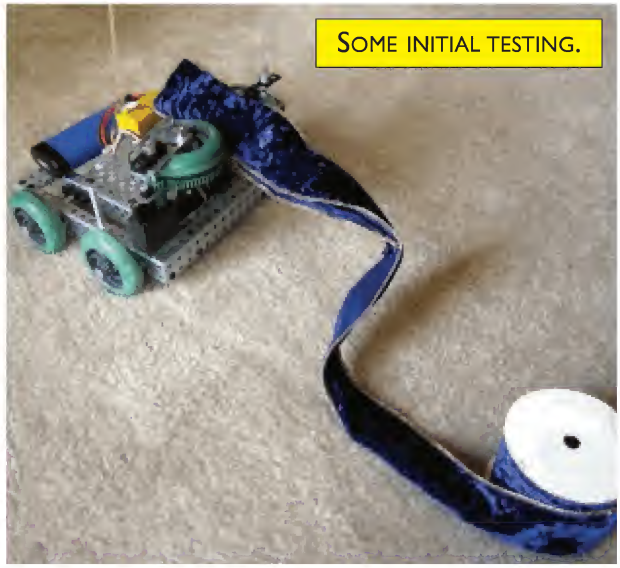
THE FIRST DRAFT OF OUR RIBBON CLIMBER.

Ribbon Dancer

A challenge like ribbon climbing may be daunting to the uninitiated, but as with any challenge the key is to break it down into manageable steps. A key part of the ribbon climber that we were keen to tackle given our background was the climbing mechanism itself. Unlike some of the other challenges inherent in a ribbon climber, the climbing mechanism appears to be rather straightforward.

Our initial design inclination was to make something akin to a softball shooter. We drew some inspiration from Deltamo,


THE CLIMBER MECHANISM.


SOME INITIAL TESTING.

Team 1079's robot from the 2006 FIRST game Aim High. Part of the challenge of Aim High was to shoot balls through a high-mounted target.

Team 1079 created a delightfully effective shooter by powering two wheels by pulley arrangements and spacing them apart so that the foam ball would be squeezed between the wheels. We geared the wheels for speed and when ushered between the wheels by a feeder mechanism, the balls would shoot out at speeds that would make them hard to avoid in a game of dodgeball. In the same way the wheels could grip the ball to shoot it, a similar mechanism could grip a ribbon and climb it.

Unlike a softball shooter where attaining a high speed for the wheels is the goal, with a climber we were far more interested in torque. A high torque mechanism would allow the climber to hoist itself up the ribbon, and the resulting lower speed of the wheels would also make for a more controlled ascent.

With this design in mind, we determined that the VEX kit would be a suitable starting point. The kit had the gears needed for a good ratio, versatile and easy to use frame components, and intuitive controls. We also thought the VEX kit would present a particularly nice challenge with respect to ribbon climbing because the parts were not exactly ideal.

With the RoboGames event, weight was at a premium — the maximum weight was about 2.2 pounds, and putting a score on the board was contingent upon not damaging the tape ribbon. The VEX kit — with its metal frame, bulky brain, and weighty wheels — is not exactly the portrait of a weight conscious bot. The extra weight, however, would make a good challenge because it meant we would have to do an even better job designing a strong climbing mechanism.

The first order of business was to sort out our gear ratio. For the best torque, we would want a small input gear (the one attached to the motor) and a large output gear (the one attached to the wheel). The VEX kit has a nice assortment of gears, but there were other limiting factors we had to consider.

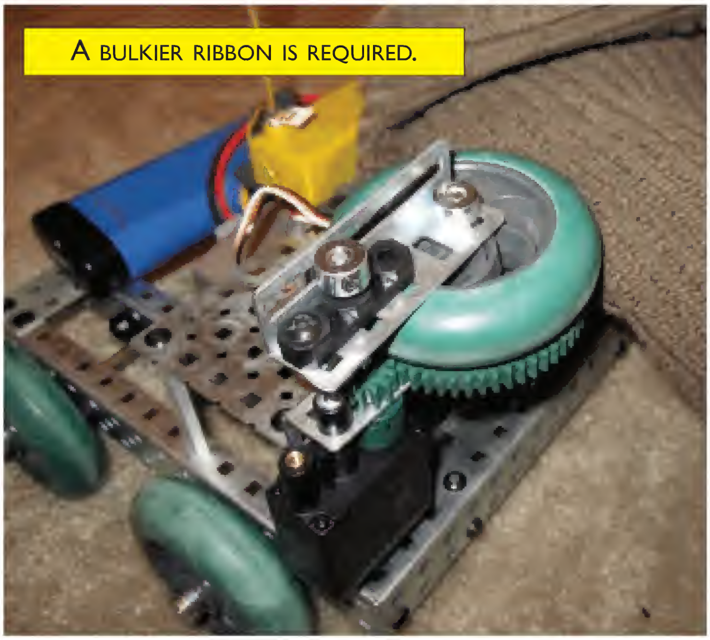
We planned on using VEX wheels for the grippers on our climbing mechanism because the rubber tires would provide great friction. Since the large output gear would be on the same axle as the gripper wheel, the size of the gear had to be slightly smaller than that of the gripper wheel. We opted to use the mid-size green VEX wheels for our preferred grippers because even though there is a larger sized wheel, the mid-size wheels were more manageable to build a bracket around while still allowing for a serious gear ratio. With the wheels chosen, we went with the gears that were the biggest possible with that pairing.

With our climbing mechanism beginning to take shape, we needed to sort out what kind of base to put it on. Many ribbon climbers at RoboGames didn't have any other mechanism aside from the climber — since the bot was only going up and down, a driving base wasn't really a necessity. We thought that attaching our climber to a driving base would add an extra layer of challenge by making it more difficult to balance the bot, and push us to make a mechanism strong enough to lift the entire assembly. We realized that adding the extra weight wouldn't align as perfectly with the goals of a RoboGames competitor, but we liked the idea of placing more emphasis on the mechanism itself.

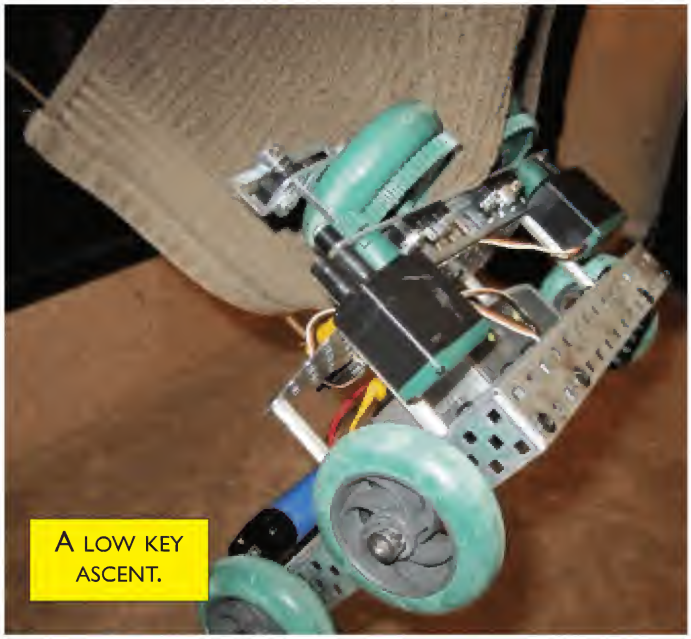
Tsiolkovsky's Eiffel Tower X110,000 may have been too heavy to be mechanically feasible, but we were confident that a driving VEX base could make the ascent.

Cliffhanger

Now that we had the design, all we had to do was everything. For the climber mechanism, we used one long frame piece as the base. Since the entire purpose of the mechanism was



A BULKIER RIBBON IS REQUIRED.



A LOW KEY ASCENT.

to grip something tightly, it would be important to give it as little play as possible. We placed the wheels so close together that they were actually in contact, and then made sure the input gears would mesh nicely with the output gears.

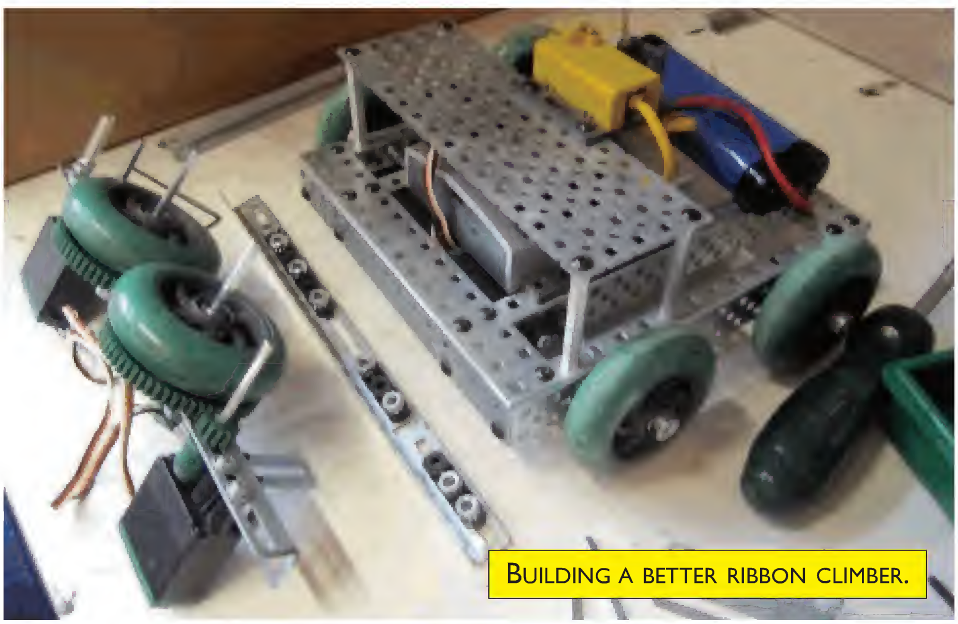
It was important to squeeze the wheels as tightly together as possible since the climbing mechanism was essentially a friction drive. Since the ribbon would have to push apart the friction wheels, this would create the normal force and the resulting friction — as long as we get the ribbon feeding through the mechanism in the first place, that is.

With everything sorted in the horizontal plane, all we had to do was tighten it up. A few more brackets allowed the climber to be attached to our driving base. We attached the mechanism to a shelf on top of the driving base, so we knew the bot wouldn't have the best balance as it hung from the ribbon. Fine-tuning the balance could be addressed once we determined if we were on the right track with our climbing mechanism.

For our initial test, we just wanted to see if the robot would devour a mock tether like a well-trained Hungry Hungry Hippo. We could think of no better treat for a ribbon climber than an actual ribbon, so we selected a velvety blue variety that promised a fair amount of friction. We doubled up on the thickness of the ribbon to give the bot more to grab onto and to increase the friction.

The initial testing was promising. When positioned properly in front of the bot, the ribbon would feed through the mechanism. The resistance we felt when pulling back on the ribbon when the bot was in a sitting position suggested that the climber had a fair bit of grip, but we were skeptical as to whether it would be enough to hoist the entire bot up the ribbon. We sat the bot up so that it was facing skyward, fed in the ribbon, and hoped for the best. The ribbon fed through the mechanism like before, but instead of lifting off of the ground the bot simply spun its wheels. We tried lifting up the ribbon to see if the bot would come with it, but the ribbon slipped from the bot's grasp like sand through the fingers.

We were still confident the mechanism could work and that we would at least have a proof of concept. All we needed was a thicker ribbon to push more against the



BUILDING A BETTER RIBBON CLIMBER.

Death defying ribbon climbing!

wheels, increasing the normal force and thus increasing the friction.

After searching for a suitable tether, we settled on a terry cloth towel (which we thought would have a better coefficient of friction too). Folding over the towel gave us a very thick tether, so much that it was a bit of a chore to begin feeding it through the mechanism. The mechanism ate up the tether with an initial test on the ground floor.

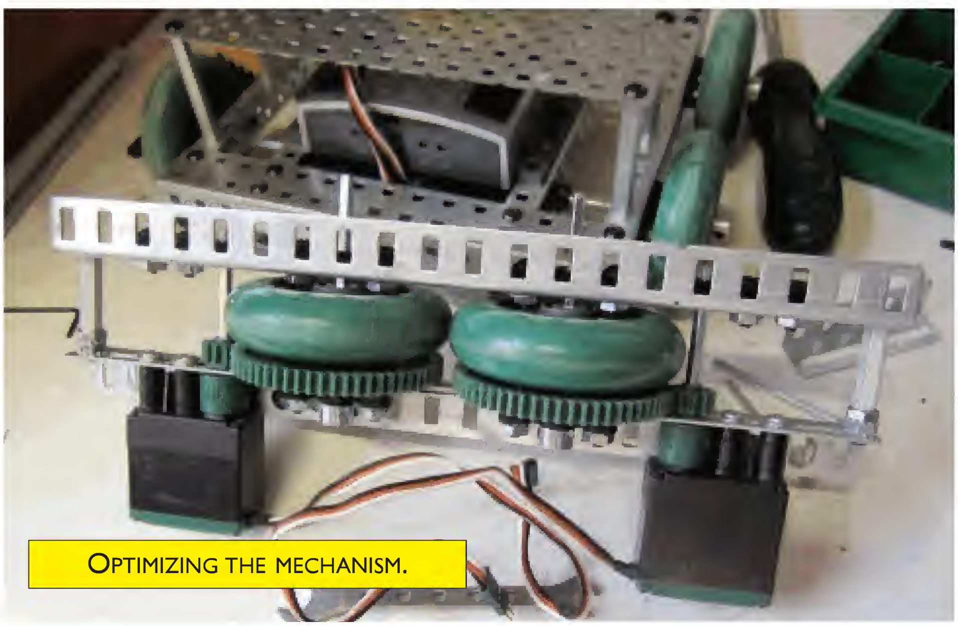
For our next trick, we sat the bot up in climbing position, had it grab a hold of the tether, and we lifted it up. Instead of sliding down the tether, the bot held fast. When we activated the climbing mechanism, the bot climbed a little bit before the tether fell out of the mechanism.

It wasn't perfect, but we had our proof of concept. We were so confident in the basic efficacy of our mechanism, that we wanted to demonstrate it in death-defying fashion. We live on the third story of our apartment building, and having the bot ascend the tether over the edge of the balcony would add the same sense of urgency as the opening scene of Cliffhanger. Fortunately for us, our ribbon climber had a stronger grip than Sly, and after a few heart-pounding seconds we brought the bot back to safety.

Optimize Prime

We were pleased that our mechanism was functional, but there was still a lot of room for improvement. The climbing mechanism thus presented a golden opportunity for a process that is of critical importance to any successful engineering project — optimization.

Optimization is the process of improving the performance of your project in a systematic way. You pinpoint specific problems and solve them. Optimization is what you do after getting the basic project working, and it's what makes the difference between something that is functional and something that is truly elegant and effective.



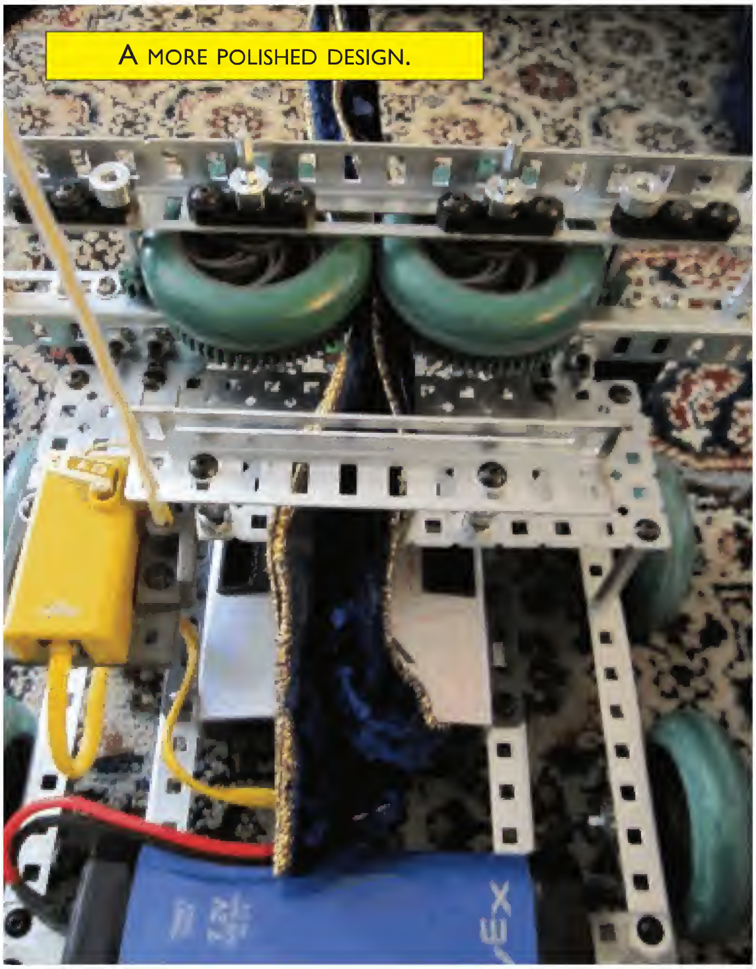
Optimizing the mechanism.

Our ribbon climber may have been functional enough to survive a cliffhanger, but relying on a thick towel and having your ascent cut short by losing your grip on the tether meant that it wasn't great. We didn't want something merely functional, we wanted something as super effective as a a Charizard's flamethrower unleashed against a grass-type.

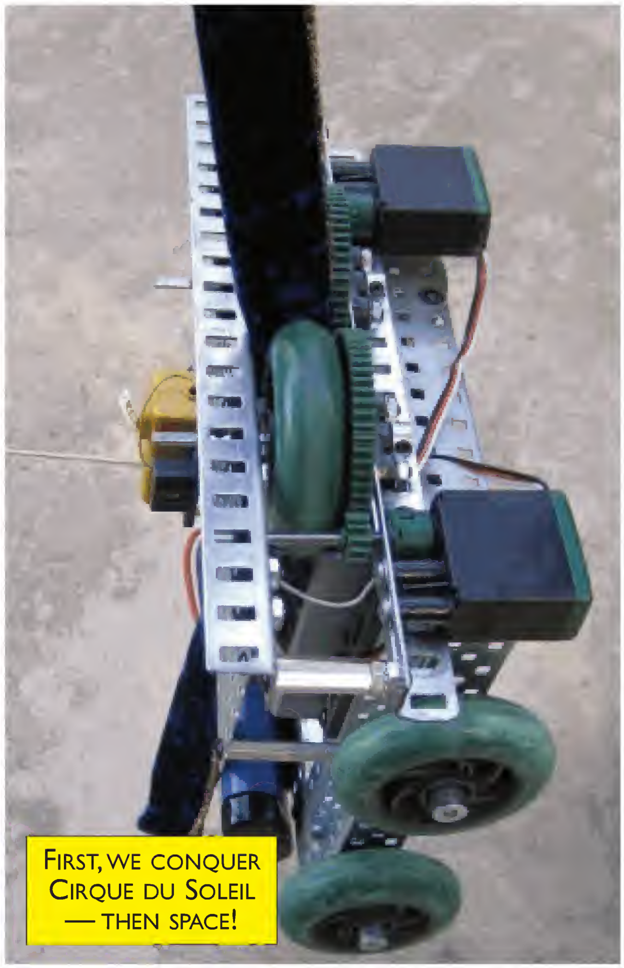
One of the major problems we noted with the first draft of our mechanism was that the wheel assemblies lacked rigidity, and so they would flex and lose grip on the ribbon. A solution to this problem was to redesign the bracket mechanism for the wheels. After rummaging through some of our other VEX parts, we found some long frame pieces that would allow us to create a more rigid and overall simpler structure. Our redesigned mechanism certainly worked better. Before, the climber could only get enough of a grip on a thick folded up towel to make any sort of ascent. With the new rigid bracket, the climber was able to ascend the much thinner blue ribbon.

Another issue we noticed with the mechanism was that the ribbon tended to bunch up and fall from above the wheels. One possible solution we imagined was to add a bracket behind the climber to feed the ribbon through as the robot ate it up. The bracket would help add some tension to the ribbon and hopefully avoid the bunching problem.

Upon initial testing, it did not look like the bracket did much of anything, but then we realized that the reason was because the ribbon lacked tension below the climber. When we tensioned the ribbon by holding down the bottom end, the bracket worked like a charm, allowing



A more polished design.



First, we conquer Cirque du Soleil — then space!

the robot to ascend without losing its grip on the ribbon.

The last optimization we thought to do was to improve the balance of the robot. Even with the bracket and improved climber, the robot could still sometimes lose its grip because the center of gravity was not in line with the ribbon.

We rearranged the mounting bracket to place it low and hanging off the front of the robot. It was not so low that the bot couldn't drive around, and it did significantly improve the bot's balance and put its center of gravity in line with the ribbon. With these improvements in place, the robot was able to ascend and descend the ribbon with ease.

Elevator Music

This project demonstrates the major benefits that can come from even simple optimizations. We realize that many projects — especially those for competition — do not always have enough time for extensive trial and error, but even some very rudimentary real world testing can reveal problems that might not have been apparent on the drawing board.

We think this project has another important lesson. Esoteric challenges like ribbon climbing can be pretty daunting. Sometimes when you try to work out something completely on paper before physically putting it together, it might seem like an endless list of design decisions and obstacles threaten to confine your ideas to the four corners of the page.

We think that sometimes just going ahead and building something can be the best way to advance a project. We don't mean to minimize the value of sketching things out and crunching the numbers, but a successful proof of concept can really jump-start a project. Nothing quite compares to the inspiration and excitement of actually building something. Once you do, you'll see solutions and new designs that may not have jumped out at you from pencil and paper.

We think this project is a great example of that. Few things seem more esoteric than a space elevator, but with a basic VEX kit and some good old fashioned trial and error, we came up with a surprisingly effective mechanism that likely isn't that far off from what you might see at future events. This isn't the sort of thing that's merely confined to competitions — there are truly exciting real world applications out there.

So, put any self-doubt aside and just build it. You might even help Tsiolkovsky's dream come true. SV

Special Thanks To

Lida

by Tom Carroll
TWCarroll@aol.com

Bases and Structure for Mobile Robots

Robot bases and their required structure are very critical parts of mobile robot design. In most cases, these vital components allow the robot to perform more complex functions such as exploring its environment. I know of many great robot builders who can write bug-proof code in their sleep but literally cringe at the task of designing a base and structure for their mobile robot. Cutting out base parts, mounting motors, and laying out sensor locations frightens many people, but these parts can be made with basic tools. You can cut sheet stock to fabricate a robot base or structure, or you can actually construct a base from beams and parts such as the articulated robot base shown in Figure 1 built with ActoBotics™ components.

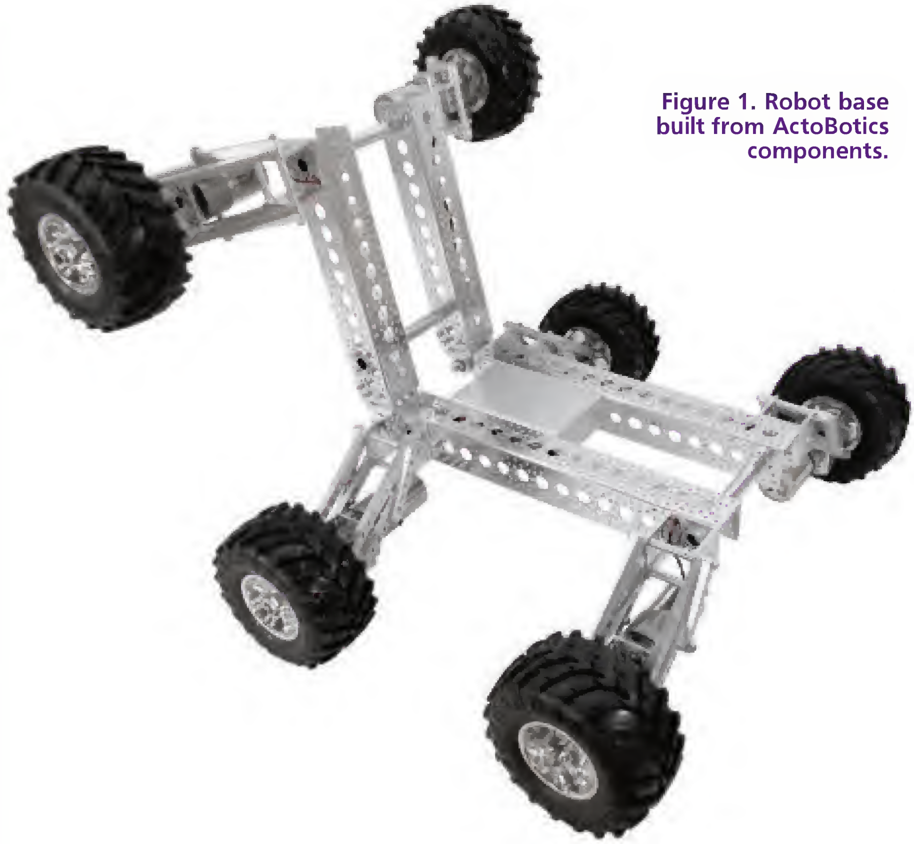


Figure 1. Robot base built from ActoBotics components.

All robot components rely on a mobile robot's base; a good design need not be complex. I wrote an article for *SERVO* discussing robot mobility about four years ago, however, some new and affordable ready-made robot bases have come on the market that I would like to discuss, as well as offer some construction tips. I'll discuss what I believe are some important facts to consider, but you can always go to the Internet for more information. I'm not going to cover aerial or underwater vehicles as these categories have their own special design considerations.

Design Issues to Contemplate

Quite a few configurations and wheel arrangements are utilized in mobile robots. The two most popular wheel/steering configurations used in mobile robot bases are Ackermann steering and differential steering.

Ackermann steering is what is called automobile type steering — two wheels in the front (or back) that steer right or left in order to change the direction of the vehicle. I will discuss this type more thoroughly later as it is not as common for mobile robots.

Post comments on this article and find any associated files and/or downloads at **www.servomagazine.com/index.php/magazine/article/december2013_ThenNow**.

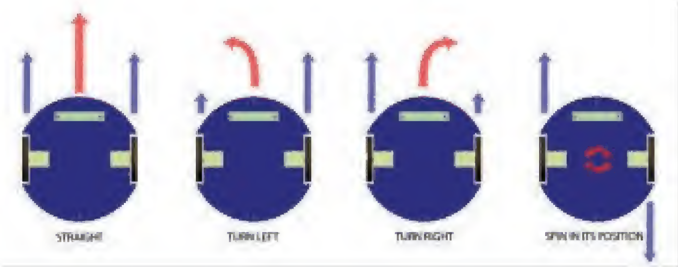


Figure 2. Differential steering from Robotix.in site.

Figure 3. Parallax HDPE wheels for the Arlo robot.

Differential steering — also known as tank-style or skid steering — utilizes different wheel speeds on each side of the vehicle to turn the vehicle; take a look at **Figure 2**. This type of steering is popular with many robot bases. If the wheel (or tank-style tread) on the right moves faster than the one on the left, the robot will turn to the left, and vice versa.

Drive Train and Power Considerations

Wheel and drive motor mounting issues can be obstacles in mobile robot design. There are so many different ways in which to mount gearmotors to robot bases that I will point out some key issues and refer you to the Internet for further information.

Reliable mounting of wheels to servos or larger gearmotors must take into consideration the intended weight of the robot and possible slopes or uneven ground to cover. Small robots weighing just a few ounces can use wheels directly mounted on small servos.

Smaller gearmotors and servos frequently have only a single bearing or bushing on the output spur gear. However, robots of several pounds or more require two bearings on the output shaft of the gearmotor in order to handle the downward bending movement on the shaft caused by the robot's weight and fore and aft bending as it travels.

Figure 3 shows a set of high quality HDPE (high density polyethylene) pneumatic wheels with a gearmotor, quadrature encoder, and mount from Parallax that is ready to attach to a larger robot.

Intended use, the available power source, the construction budget, the builder's mechanical skill level, and even aesthetics are all important issues to consider. If your desire is for an SRS (Seattle Robotics Society) RoboMagellan robot, you might want to look at an off-road RC car chassis with Ackermann steering to cover a rough outdoor environment.

If you are designing a tall indoor robot to serve you or guests, you would look at differential steering for quick turns and a wide base for stability (to prevent tipping over). Most small kit robots also utilize differential steering. These are just two examples of many designs to consider; I'll delve into the two steering types a bit more later.

Robot Base Structural Materials

Robots can be made of almost any material, though many hobbyist-built robots end up being constructed out of aluminum. My first robot (years ago) was made from a large gallon size 'tin' can. A later one I designed for *Boys Life Magazine* was made from a plastic trash can.

The largest robot that I built as a high school student was made from 3/4" plywood with jukebox motors and parts for the mechanisms. It was a huge beast, and I would never recommend anyone making a robot from such heavy materials.

Wood, however, is a great material — especially thinner types of craft plywood and even cabinetry wood. Wood is easy to cut, trim, drill, and sand, and it is relatively cheap and readily available. **Figure 4** shows a prototype of the old MadeUSA base from Parallax that is made from a high grade furniture plywood. I highly recommend using screws rather than

Figure 4. Parallax robot with plywood base.

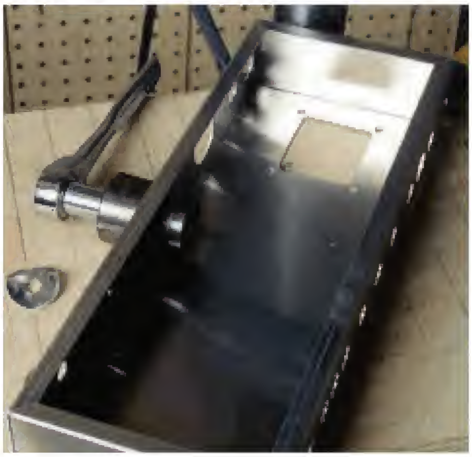


Figure 5. Chassis punch in use.

Figure 7. Mark Curry's Nomad in 2012 SRS RoboMagellan.

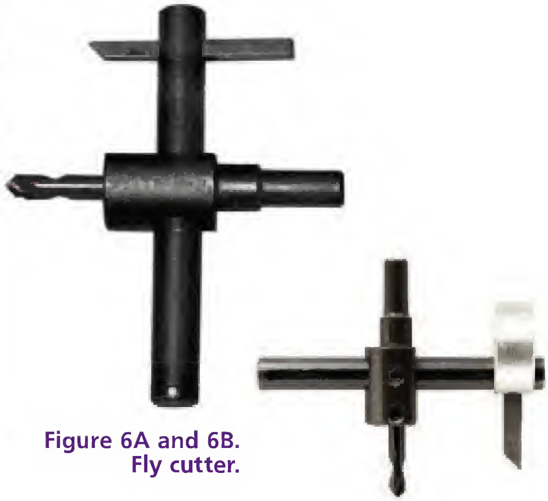


Figure 6A and 6B. Fly cutter.

welding (as is typical for assembling a metal robot) since screws can be removed or placed differently to 'tweak' a robot's chassis — welds cannot.

Aluminum sheet metal — especially 6061-T6 stock — is, in my opinion, the best material for a robot base, though I have also used harder 7075 in applications where little machining is required. Both types are readily available at many locations and are easy to cut with a bandsaw, though I would suggest using a large metal shear at a nearby machine shop or technical college as the edges are so much smoother (especially for thicker stock). Small holes are easily cut into thinner aluminum and steel stock using a chassis punch as shown in **Figure 5**. Drill a small pilot hole and use a ratchet wrench to tighten the three punch pieces until the hole pops through.

Cutting large holes in any sheet metal is a bit harder. Large bandsaws have a way to cut the blade and insert it in a small pilot hole, and then weld it back together with a welder mounted on the saw. If you don't have access to one of these, you can use a Sawzall type of reciprocating saw or hand jigsaw fitted with a metal cutting blade to enlarge holes for wheels and such.

It takes a bit of practice to cut a nice hole. You'll need to do a bit of filing to smooth out the ragged edges. A fly cutter as shown in **Figure 6** can cut larger round holes using a drill press.

Make sure your work piece is securely clamped down. Either of these methods work for mild sheet steel, as well. Plywood, HDPE, most plastics, brass stock, and just plain lumber can also be used for robot bases and structure.

Designing Robots for Outdoor Operation

Considering the growing popularity of the RoboMagellan contests around the country, I will start with base designs that are best for these types of autonomous robots that must traverse rough outdoor courses during competitions.

I was just at the SRS's 2013 Robothon exposition at the Seattle Center (in Seattle) and witnessed another RoboMagellan contest. An off-road configuration RC car chassis like Mark Curry's robot in the 2012 SRS RoboMagellan competition shown in **Figure 7** is typical of most entrants. Large knobby tires coupled with a beefy drive train mounted on a rigid base can easily traverse thick grass and inclines.

A large hobby shop — especially one that deals mostly in off-road RC cars — will have almost any type of car chassis you might want for your small to mid-size outdoor robot. You may opt for purchasing the car from an Internet seller, but I strongly recommend that you personally handle and inspect one in a store just to see how it is constructed and applicable to your project.

Remember, it is not going to be an RC vehicle anymore; you are going to need special power supply setups for the motor drivers and also for the other electronics. Sensors (cameras and object detection), GPS antenna mounting, shaft encoders, and one or more microcontrollers need unique mounting locations.

If you are mostly interested in RoboMagellan-style robots that generally operate at lower speeds, large shock-absorber mounted wheels are not that important, but that also depends on the area where the competitions are to be held.

At this year's Robothon, one RoboMagellan contestant's entry was doing quite nicely until it happened upon a steep 6" edge of grass next to a sidewalk. Instead of smoothly transitioning from the grass to the pavement, the robot dropped at a 45º angle to the edge of the concrete and stalled, unable to move forward or backward.

Look for vehicles that have wheels that protrude past the front and rear portions of the chassis to allow them to grip and climb over obstacles. Remember that larger wheels may have more traction surface, but

require a lower gear speed with greater torque for the same traveling speed.

RoboMagellan robots are just one of many varieties of mobile machines that operate outdoors and need some special design considerations. A roving video platform, an autonomous lawn mower, a cave explorer, or even a dog-walking robot can make for an interesting project.

Needless to say, you will probably encounter dirt, grass, and moisture along the way, so you need to plan a way of sealing and protecting electronics, encoders, motors and gears, batteries, and sensors.

Ackermann Steering for the Robot's Base

Let's discuss Ackermann steering next (refer to **Figure 8**). Notice that the pivot arms on each of the front wheels are not parallel with the sides of the wheels, but actually form a 'V' with lines extended to a point centered between the rear wheels as shown in the left drawing. This is called the Ackermann angle. The drawing on the right shows how the two front wheels are not parallel when turned to an extreme to allow a single turning point for all four wheels.

As you can see, only one servo would be required to turn both wheels, but the pivot arms must be set at the proper angle when connected to the cross bar. Many ready-built RC cars chassis' will already be configured this way, unless you change the wheel base for your robot.

This style of steering requires an entirely different type of control than many robot builders are used to. Your robot will require a steering motor/servo and one or two driving motors for the rear wheels. Most robot builders place shaft encoders on the steered wheels for feedback as they just 'skate' across the ground without any torque skidding.

If the base is a true 4WD setup, the encoders can be placed on the rear. You should not use a single rear shaft connecting the rear wheels to a single motor, but rather use a differential between the two wheels like those used in automobiles. Since these types of gear arrangements are difficult to build or expensive to buy for a hobbyist, most robot builders use two separate motors — one for each rear wheel.

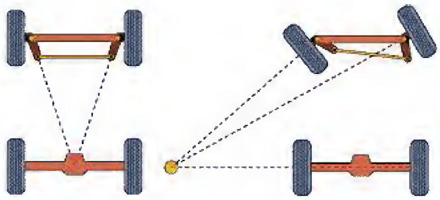


Figure 8. Ackermann steering.

Figure 9 shows a set of front and rear wheel axle/steering assemblies from an online supplier that uses a single motor and a differential. In most cases, one motor will operate either faster or slower as required when equal drive voltages are applied to both motors, without skidding one or both wheels.

Not all off-road RC car chassis configurations utilize Ackermann steering. Many use four or six wheel designs, with each side motor sets of two or three wheels being driven together to form a differential driving configuration. **Figure 10** shows an arrangement of six sets of servo-driven wheels — each with another servo mounted on top to a plastic base plate to individually steer each one.

Figure 9. Ready-built Ackermann steering for RC cars.

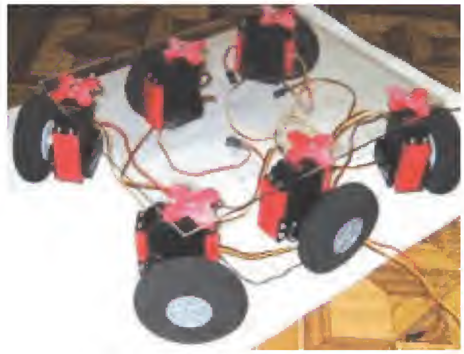


Figure 10. Six-wheel drive with individual six-wheel steering.

The Mars Curiosity rover is an example of a combination of six individually steered wheels that allow the robot to steer 360º about its center. **Figure 11** shows Curiosity in a lab setup with its left-front wheel turned at an angle. Notice that the rotating shaft is directly above the center of turning rotation. This one ton robot is powered by a plutonium

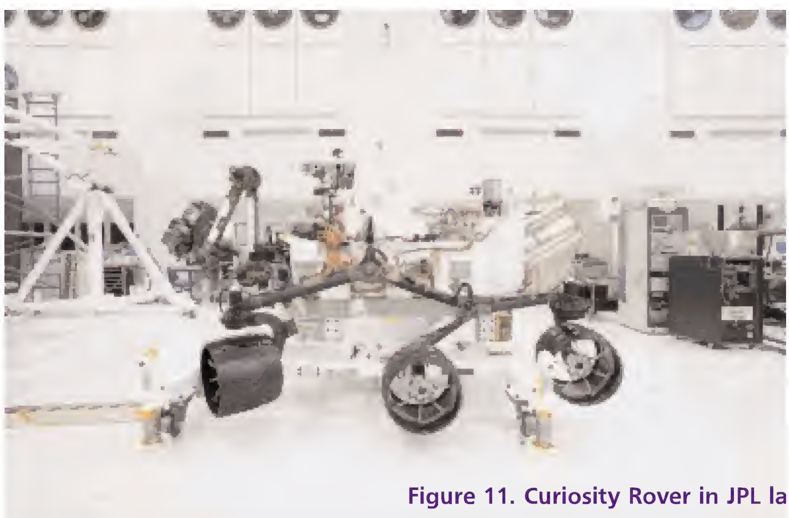


Figure 11. Curiosity Rover in JPL lab setup.

Figure 12. Odyssey 6WD robot kit from the RobotShop.

powered RTG (Radioisotope Thermoelectric Generator) that only delivers 125 watts of electric power, so no friction-causing skids can be tolerated.

A less complex arrangement is the Odyssey robot kit from RobotShop shown in **Figure 12.** Note that all six wheels are mounted in a straight line, and you can see the individual brackets on each motor/wheel that allow each wheel to swivel up and down separately from the others. This is good for straight forward driving since all six wheels can generate a lot of forward force on the ground.

The negative aspect is that it forms a skid-steering configuration that is inefficient and takes more power from your battery than the true Ackermann configuration. I'll discuss another 6WD mid-size robot base that uses this configuration in the ready-built robots later.

Indoor Robot Design Considerations

Designing a robot for use inside has its own advantages — rain, moisture, and large amounts of dirt are usually not a problem. Differential steering is frequently used due to the turning ability mentioned earlier. It requires a different type of programming since two driving motors are used for all directional movement and turning.

Movement can be accomplished by driving each of two side wheels at different speeds and directions as seen in the earlier **Figure 2**. The robot will require a freely-moving front or rear swivel caster (or even two casters) to stabilize the robot. If the robot is to be a two-wheel 'Segway' style self-balancing robot, you still might want to incorporate one or two casters or extendible 'feet' to catch the robot should it topple over.

A self-balancing robot looks cool, but I have seen more than one of these types of robots smash a video display, iPad, or camera when falling on its face.

Telepresence robots are becoming very popular, and they require height. Robot height places stress on bases and wheels so that the wheel base should be extended as far as possible — both to the sides, and fore and aft — to prevent falling over. The casters should not be too firmly attached so that they both touch the floor at the same level as the drive wheels.

If the robot happens to drive over a door's threshold and stops with the caster wheel atop the threshold, the drive wheels may not be able to touch the floor to continue driving. Use spring-loaded mounts for the casters that are stiff enough to prevent the robot from rocking, yet not too stiff to prevent the robot's weight from keeping the main drive wheels on the floor.

Keep heavy items such as batteries as low as possible — especially heavier lead-acid batteries. Speaking of batteries, it is very tempting to use Lithium Polymer (LiPo) batteries for their lightness and power density but take a look at these two photographs: **Figure 13** is of David Anderson's nBot just after completion; **Figure 14** shows the robot after the LiPo batteries (seen in yellow) decided to just burn up when the robot was dormant and take the robot with them. He rebuilt the robot using NiMH batteries. LiPos caused the new Boeing Dreamliner a lot of scary moments until Boeing found a way to seal them and vent any gasses to the outside.

That is hard to do in a small robot, so my advice is to charge them correctly

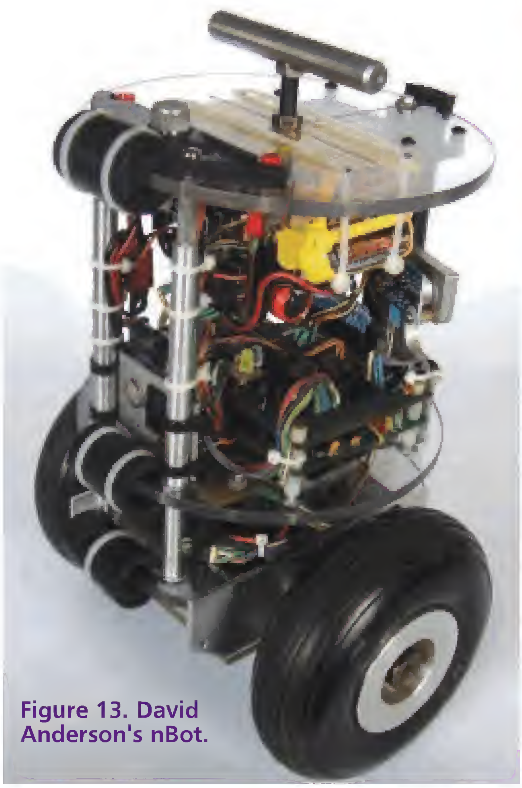


Figure 13. David Anderson's nBot.

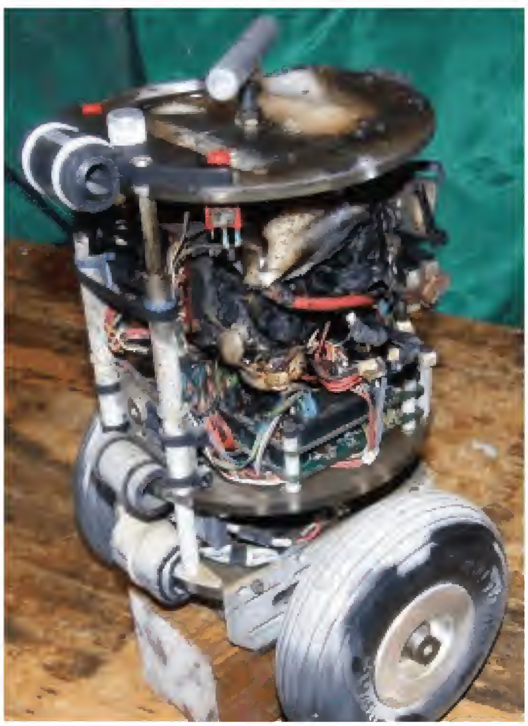


Figure 14. David Anderson's nBot destroyed by LiPo fire.

Figure 15. Original Parallax MadeUSA base with Eddie controller.

Figure 16. New Parallax Arlo robot with HDPE wheels and mounts.

outside the robot with LiPo chargers and use them wisely. Store them outside your robot when not in use, as they can ignite when being charged or when the robot is in a dormant state.

Model helicopters, quadcopters, and robots with LiPo batteries have set many car's backseats on fire. That's the price you pay for otherwise great batteries.

Ready-built Robot Bases

Parallax built a large 18" diameter robot base for their Eddie robot, and also for their MadeUSA (see **Figure 15**). This particular photo shows an added Eddie controller board seen on the bottom platform. It was a bit pricey for most builders as it had expensive-to-make machined wheels and mounts made in the US-based Parallax facility, and 10 Ping))) and IR sensors on the periphery. I've consulted with Ken Gracey and his engineers on the design of a low cost yet high quality large base and they have come up with a $336 base called the Arlo Robot. (They are hoping to drop the price even lower.)

The new base uses the same gearmotors and 18" platforms, but uses molded plastic mounts and wheels as shown in **Figure 16**. Adding their HB-25 fan-cooled motor controllers, a second platform, IR and ultrasonic sensors, quadrature encoders, and even using machined aluminum wheels and mounts are available as upgrades or options.

The Wild Thumper robot base shown in **Figure 17** is an ideal mid-size base at 16.5" long by 11.5" wide at the wheels, and 5" high. It weighs six pounds and has a 5 kg/11 pound payload capacity. It features a 2.5" ground clearance when lightly loaded.

Several robot suppliers carry this base and some add their own drivers. The $350 Parallax version includes two of the Parallax HB-25 fan-cooled motor controllers and also a Propeller project board. The six wheels are independently spring-loaded to traverse rough terrain but do have Ackermann steering.

To drive the robot, the three right motors and the three left motors are driven differentially by the two motor controllers. It is definitely skid steering. Drawing 6.6 amps each at stall, that's 19.6 amps for each side. Internally, there is room for the two controllers in one compartment and a LiPo or NiMH battery in the other

Figure 17. Wild Thumper six--wheel drive robot base.

compartment. You can drive the wheels at 350 RPM for a 34:1 ratio gearmotor — that is a fairly high speed for a robot. It is one great base for the price.

iRobot came out with the Create shown in **Figure 18** in 2007 as a way

Figure 18. iRobot Create base.

Figure 19. Kobuki mobile robot base.

Figure 20. Pololu Arduino robot kit.

for hackers to have a tried-and-proven robot base without tearing up a Roomba vacuum cleaner. This platform is way better for the experimenter as it contains 30 built-in sensors, a 25-pin expansion port for connecting a command module and other electronics, and a spacious cargo bay with plenty of holes for mounting hardware. Retail is $130 for the basic model or $220 for an added advanced rechargeable battery pack and fast charger.

iRobot no longer offers the Command module, but there is so much Create hacking information on the Internet that any Arduino, Propeller, or AVR microcontroller is easily used for experimentation.

In 2011, Yujin Robot of South Korea developed the Kobuki shown in **Figure 19** which was modeled after their Iclebo robot home vacuum cleaner. It claims to have corrected features with higher resolution wheel encoders, built-in gyro, and an available larger battery. Willow Garage — maker of the ROS-powered TurtleBot — originally used the Create with their robot, then changed over to the Kobuki. nfortunately, sales have not been very successful.

I personally prefer the Create for their support, but a Kobuki-based TurtleBot can be found on the Internet for $1,400 that includes mounting plates, cables, and a Kinect.

The Pololu Arduino robot at $275 (shown in **Figure 20**) is a great small experimenter's robot platform. It has two ATmega32U4 microcontrollers (including a 2.5 KB SRAM and 1 KB EEPROM), a 160x120 pixel color LCD, a speaker, digital compass, SD card reader, four prototyping areas, and an SD card preloaded with images and sound files used in example Arduino robot projects.

The Parallax Boe-Bot — starting at $170 and shown in **Figure 21** — is a perfect small experimenter's platform. The basic model uses Parallax's popular BASIC Stamp microcontroller, but other versions can be configured with their Propeller eight-cog microcontroller, or the Arduino. You can mount ultrasonic distance detectors, X-Bee RF links, speech generators, optical detectors, and other items to Boe-Bots.

Parallax has an extensive course library to lead a beginner or even an expert through programming and sensor technology to industrial applications.

All six of the above robot bases and kits are excellent ways for someone new to robotics to experiment and find their ultimate robot platform. Buy or build — it's your choice.

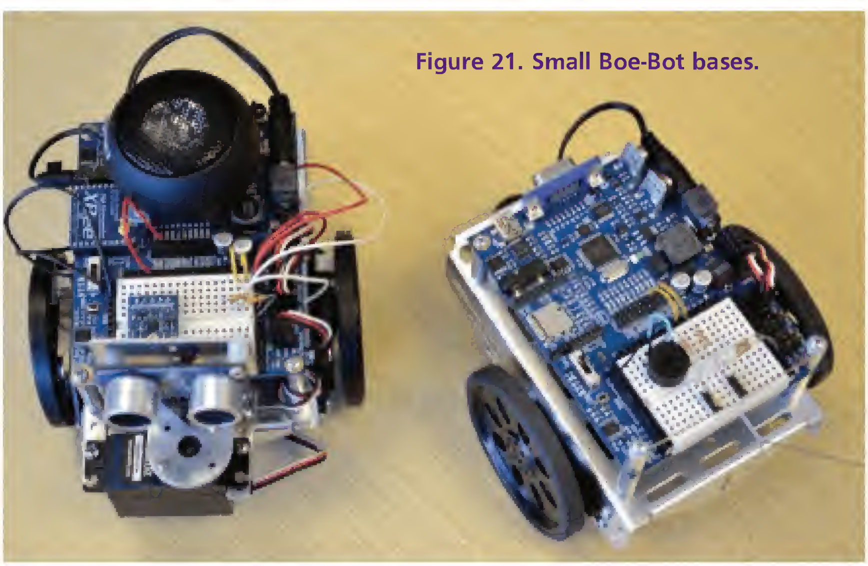


Figure 21. Small Boe-Bot bases.

Final Thoughts

This article was not intended to teach machine shop techniques, but to toss out some important factors to assist robot builders in designing and building their robot. If you are new to robot experimentation and feel that you need some help in designing structures and a base for your prospective robot project, consider joining a robot club. Many robotics organizations have a website that allows members and outsiders to ask questions, showcase projects, and offer help to newcomers. A neighbor or friend who is mechanically adept does not need to know about robots in order to assist you in designing basic parts. You can start with one of the robot bases described here and build your dream robot upon it. Simplicity in construction, readily-available bases, and friendly assistance from other builders will lead you to a great robot. **SV**

ROBO-LINKS

Advertiser Index

SERVOCITY
sales@servocity.com
(620) 221-0123
MAKING THINGS MOVE
LINEAR ACTUATORS
12V HEAVY DUTY
Up to 180 lb thrust
Offered in 6 stroke lengths from 2" to 12"
Aluminum stroke rod and outer tube
Metal drive gears ensure tremendous durability
5/16" diameter mounting points that bolt up to our mounting brackets
(sold separately)
$129.99
Limit switches at both ends to prevent over-run
Built-in potentiometer can be easily accessed for position feedback
Easily integrate an actuator into your Actobotics™ project!
12V SUPER DUTY
Up to 1,570 lb thrust
Offered in 6 stroke lengths from 4" to 24"
Powder-coated aluminum base
1.125" diameter extension shaft
Precision ball screw-drive
$399.99
Stainless steel extension rod and steel outer tube
Super-duty mounts available (sold separately)
GEARMOTORS
3-12V Standard Gearmotors
$24.99
6-12V Micro GearmotorBlocks
$14.99
$59.99
Robotzone Heavy Duty Gearmotors
6-12V Precision Gearmotors
$49.99
Offset Gearmotors
$34.99
MOTOR SYSTEMS
These systems offer a simple way to reduce the speed and increase the torque of a gearmotor
$59.99
Gearmotor Gearbox
(full metal gears)
$159.99
Belt Drive Pan System
$59.99
Worm Drive Gearbox
30:1 Gear Reduction Ratio
HS-7955TG
$119.99
HSR-5980SG
$109.99
83.3 oz-in of torque!
$16.99
HS-485HB
1/4 Scale Servo
$39.99
HS-755MG
$69.99
HS-7235MH
SERVOS
HiTEC
— DID YOU KNOW? —
We price match and pre-modify Hitec servos!
Brand New!
Actobotics
BALL-BEARING PRECISION BUILDING SYSTEM
DREAM.
DESIGN.
BUILD.
REPEAT.
SCAN THIS CODE TO VISIT SERVOCITY ON THE GO!
Rover Build Credit:
John Artt
Robotics Teacher
Ellensburg, WA
Channel & Brackets
Hub Mounts
Bearing Mounts
Gears & Sprockets
Shafting & Tubing
Hubs & Clamps
Channel Gearboxes
Wheels & Tires
Pulleys & Belts
...and much more!
WE SHIP WORLDWIDE
One online store.
Thousands of products.
Millions of possibilities.
www.SERVOCITY.com
HUGE SELECTION - LOW PRICES - FAST SHIPPING!
Join our online community